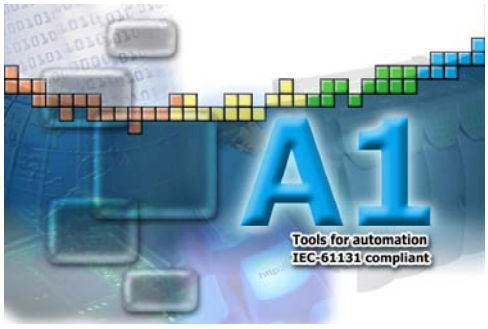
Manual de Instruções

A1 Automation Tools

Manual Rev. 1.30 Maio / 2007 **Ref.2-086.130**



Este manual não pode ser reproduzido, total ou parcialmente, sem autorização por escrito da Atos.

Seu conteúdo tem caráter exclusivamente técnico/informativo e a Atos se reserva no direito, sem qualquer aviso prévio, de alterar as informações deste documento.

Termo de Garantia

A **Atos Automação Industrial LTDA.** assegura ao comprador deste produto, garantia contra qualquer defeito de material ou de fabricação, que nele apresentar no prazo de 360 dias contados a partir da emissão da nota fiscal de venda.

A **Atos Automação Industrial LTDA.** restringe sua responsabilidade à substituição de peças defeituosas, desde que o critério de seu Departamento de Assistência Técnica, se constate falha em condições normais de uso. A garantia não inclui a troca gratuita de peças ou acessórios que se desgastem naturalmente com o uso, cabos, chaves, conectores externos e relés. A garantia também não inclui fusíveis, baterias e memórias regraváveis tipo EPROM.

A **Atos Automação Industrial LTDA.** declara a garantia nula e sem efeito se este produto sofrer qualquer dano provocado por acidentes, agentes da natureza, uso em desacordo com o manual de instruções, ou por ter sido ligado à rede elétrica imprópria, sujeita a flutuações excessivas, ou com interferência eletromagnética acima das especificações deste produto. A garantia será nula se o equipamento apresentar sinais de ter sido consertado por pessoa não habilitada e se houver remoção e/ou alteração do número de série ou etiqueta de identificação.

A **Atos Automação Industrial LTDA.** somente obriga-se a prestar os serviços referidos neste termo de garantia em sua sede em São Paulo - SP, portanto, compradores estabelecidos em outras localidades serão os únicos responsáveis pelas despesas e riscos de transportes (ida e volta).

• Serviço de Suporte Atos

A **Atos** conta com uma equipe de engenheiros e representantes treinados na própria fábrica e oferece a seus clientes um sistema de trabalho em parceria para especificar, configurar e desenvolver software usuário e soluções em automação e presta serviços de aplicações e startup.

A **Atos** mantém ainda o serviço de assistência técnica em toda a sua linha de produtos, que é prestado em suas instalações.

Com o objetivo de criar um canal de comunicação entre a **Atos** e seus usuários, criamos um serviço denominado **Central de Atendimento Técnico**. Este serviço centraliza as eventuais dúvidas e sugestões, visando a excelência dos produtos e serviços comercializados pela **Atos**.

Central de Atendimento Técnico
De Segunda a Sexta-feira
Das 7:30 às 12:00 h e das 13:00 às 17:30 h
Telefone: 55 11 5547 7411
E-mail: suportec@atos.com.br

CONVENÇÕES UTILIZADAS

• Títulos de capítulos estão destacados no índice e aparecem no cabeçalho das páginas;

• Palavras em outras línguas são apresentadas em *itálico*, porém algumas palavras são empregadas livremente por causa de sua generalidade e freqüência de uso. Como, por exemplo, às palavras software e hardware.

Números seguidos da letra h subscrita (ex:1024_h) indicam numeração hexadecimal e seguidos da letra b (ex:10_b), binário. Qualquer outra numeração presente deve ser interpretada em decimal.

• O destaque de algumas informações é dado através de ícones localizados sempre à esquerda da página. Cada um destes ícones caracteriza um tipo de informação diferente, sendo alguns considerados somente com caráter informativo e outros de extrema importância e cuidado. Eles estão identificados mais abaixo:

NOTA: De caráter informativo, mostra dicas de utilização e/ou configuração possíveis, ou ressalta alguma informação relevante no equipamento.

OBSERVAÇÃO: De caráter informativo, mostra alguns pontos importantes no comportamento / utilização ou configuração do equipamento. Ressalta tópicos necessários para a correta abrangência do conteúdo deste manual.

IMPORTANTE: De caráter informativo, mostrando pontos e trechos importantes do manual. Sempre observe e analise bem o conteúdo das informações que são identificadas por este ícone.

ATENÇÃO: Este ícone identifica tópicos que devem ser lidos com extrema atenção, pois afetam no correto funcionamento do equipamento em questão, podendo até causar danos à máquina / processo, ou mesmo ao operador, se não forem observados e obedecidos.

Conteúdo

CAPÍTULO 1 9

INSTRUÇÕES 9

Conjunto de Instruções 11

Aritméticas 11
Array 11
Biestáveis 11
Comparação 11
Conversão 12
Data/hora 12
Detecção de borda 12
Logarítmicas 12
Lógica entre registros 13
Movimentação de dados 13
Manipulação de strings 13
Seleção 14
Temporização e contagem 14
Trigonométricas 14
Especiais 15

CAPÍTULO 2 17

INSTRUÇÕES DE LÓGICAS 17

Instruções de lógica entre contatos 19

• Contato Normalmente Aberto (NA) 19
• Contato Normalmente Fechado (NF) 19
• Contato de Transição Positiva 19
• Contato de Transição Negativa 20
• Saída 20
• Saída Invertida 20
• Set Saída 20
• Reset Saída 21
• Saída de Transição Positiva 21
• Saída de Transição Negativa 21
• Saída SET RESET 22

CAPÍTULO 3 23

INTRUÇÕES IEC61131-3 23

Aritméticas 25

ABS - Módulo 25
ADD - Adição 25
DIV - Divisão 26
EXP - Exponencial 26
EXPT - Potência X^Y 27
MOD – Módulo de uma divisão 27

MOVE - Movimentação de dados 28
MUL - Multiplicação 28
SQRT - Raiz Quadrada 29
SUB - Subtração 29

Biestáveis 30

RS – Biestável com Reset dominante 30
SR – Biestável com Set dominante 30

Comparação 31

EQ - Igual 31
GE - Maior ou igual que 31
GT - Maior que 32
LE - Menor ou igual que 32
LT - Menor que 33
NE - Diferente 33

Conversão 34

*_TO_** 34
*_TO_BCD 34
*_TO_STR – Converte para STRING 35
BCD_TO_** 35
STR_TO_** - Converte STRING para INT/DINT 35
TRUNC 36

Logarítmicas 37

LN - Logaritmo natural 37
LOG - Logaritmo de base 10 37

Lógica entre registros 38

AND 38
NOT 38
OR 39
XOR 39

Movimentação de dados 40

ROL - Rotação à esquerda 40
ROR - Rotação à direita 40
SHL - Deslocamento à esquerda 41
SHR - Deslocamento à direita 41

Manipulação de Strings 42

CONCAT - Concatenação 42
DELETE - Apagar 42
FIND - Procurar 43
INSERT - Inserir 43
LEFT – Copia n caracteres da string a partir da esquerda 44
LEN - Comprimento 44
MID - Copiar 45
REPLACE - Substituir 45
RIGHT – Copia n caracteres da string a partir da direita 46

Seleção **47**

LIMIT – Limitador de mínimo e máximo 47
MAX – Seleciona maior valor 48
MIN – Seleciona menor valor 48
MUX – Multiplexador extensível 49
SEL – Seleção binária 50

Temporização e Contagem **51**

CTD - Contador Decrescente 51
CTU - Contador Crescente 53
CTUD - Contador Crescente e Decrescente 55
TOF - Temporizador: OFF Delay 58
TON - Temporizador: ON Delay 60
TP - Temporizador: Pulse mode 62

Trigonométricas **64**

ACOS - Função ARCO COSSENO 64
ASIN - Função ARCO SENO 64
ATAN - Função ARCO TANGENTE 65
COS - Função COSSENO 65
SIN - Função SENO 66
TAN – Função TANGENTE 66

CAPÍTULO 4 67

INTRUÇÕES ATOS 67

Array **69**

XOR_AR 69
*_TO_AR 70
AR_TO_* 70

Comparação **71**

ALARM - Geração de alarmes 71

Conversão **72**

DATE_FORMAT 72
DEG_TO_RAD - Graus >> Radianos 73
RAD_TO_DEG - Radianos >> Graus 73

Data/hora **74**

RTC_ATOS – Real Time Clock 74

Detecção de borda **75**

F_TRIG – Detecta borda de descida 75
R_TRIG – Detecta borda de subida 76

Especiais **77**

ADJUST - Ajuste de escala 77
ANGLE COUNTER 78
ANGLE SIMULATE 79
CEP – Controle Estatístico do Processo 80

CEP – (Continuação) 82
CPU COUNTER - Contador da CPU 83
Jumpers de configuração 83
EVENT_LOG – Log de eventos 84
FILT - Filtro 86
Exemplo de FILT 86
HIGH SPEED COUNTER 88
PID - Algoritmo PID ISA 89
PID - Algoritmo PID ISA (continuação) 90
RECIPE – Arquivo de receitas 91
RECIPE(Continuação) 93
RPRINT – Leitura de canal serial 93
RPRINT(Continuação) 94
SCALE - Ajuste de escala 95
SCALE2G - Escala de 2º grau 96
SPRINT – Escrita em canal serial 97
STEPMOTOR_FASE – Motor de passo modo fase 98
Passo inteiro 100
Meio Passo 100
Tipo "N" - 4004.05BF (Somente para tipo N) 101
STEPMOTOR_PULSE – Motor de passo modo pulse 102
Tipo "N" - 4004.05BF 104
Tipo "P" - 4004.06BF 104

Movimentação de dados 105

BITW - Movimentação BIT >>> WORD 105
BMOVX - Movimentação indexada na origem e destino 106
MOVX - Movimentação indexada no destino 107
MOVX - Movimentação indexada no destino 108
SFR - Deslocamento 108
SCROLL – Deslocamento de bloco de dados 109
Exemplo 1 110
Exemplo 2 111
Exemplo 3 112
Exemplo 4 112
TAB - Carregamento de constantes 113
WBIT - Movimentação WORD >>> BIT 114
WBITX - Movimentação indexada WORD >>> BIT 115
WLDX - Movimentação indexada na origem 116
Exemplo de range de dados 116
Endereço Mínimo 117
Endereço Máximo 117

Temporização e Contagem 118

TMR_1MS - Temporizador: 1ms OFF Delay 118
TON_100MS - Temporizador: 100ms ON Delay 119
TON_1S - Temporizador: 1s ON Delay 120

CAPÍTULO 1
INSTRUÇÕES

Conjunto de Instruções

Esta lista compreende todas as instruções de todas as bibliotecas existentes no aplicativo A1.

Aritméticas

NOME	DESCRIÇÃO [Biblioteca da função]
ABS	• Módulo [IEC]
ADD	• Adição [IEC]
DIV	• Divisão [IEC]
EXP	• Exponencial [IEC]
EXPT	• Potência [IEC]
MOD	• Módulo de uma divisão [IEC]
MOVE	• Movimentação de dados [IEC]
MUL	• Multiplicação [IEC])
SQRT	• Raiz Quadrada [IEC]
SUB	• Subtração [IEC]

Array

NOME	DESCRIÇÃO [Biblioteca da função]
***_TO_AR**	• Adiciona dado no Array [ATOS]
AR_TO_*	• Retira dado do Array [ATOS]
XOR_AR	• XOR no array de bytes [ATOS]

Biestáveis

NOME	DESCRIÇÃO [Biblioteca da função]
RS	• Biestável com Reset dominante [IEC]
SR	• Biestável com Set dominante [IEC]

Comparação

NOME	DESCRIÇÃO [Biblioteca da função]
ALARM	• Geração de alarmes [ATOS]
EQ	• Igual [IEC]
GE	• Maior ou igual que [IEC]

GT • Maior que [IEC]

LE • Menor ou igual que [IEC]

LT • Menor que [IEC]

NE • Diferente [IEC]

Conversão

NOME **DESCRIÇÃO** [Biblioteca da função]

_TO_* • Conversão entre os tipos de dados [IEC]

***_TO_BCD** • Converte para BCD [IEC]

***_TO_STR** • Converte para STRING [IEC]

BCD_TO_** • Converte valor BCD para tipo UINT/UDINT [IEC]

DATE_FORMAT • Converte data para string formatada [ATOS]

DEG_TO_RAD • Converte graus em radianos [ATOS]

RAD_TO_DEG • Converte radianos em graus [ATOS]

STR_TO_** • Converte STRING para UINT [IEC]

TRUNC • Truncamento [IEC]

Data/hora

NOME **DESCRIÇÃO** [Biblioteca da função]

RTC_ATOS • Real Time Clock [ATOS]

Detecção de borda

NOME **DESCRIÇÃO** [Biblioteca da função]

F_TRIG • Detecta borda de descida (mudança de estado 1 para 0) [IEC]

R_TRIG • Detecta borda de subida (mudança de estado 0 para 1) [IEC]

Logarítmicas

NOME **DESCRIÇÃO** [Biblioteca da função]

LN • Logaritmo natural [IEC]

LOG • Logaritmo de base 10 [IEC]

Lógica entre registros

NOME	DESCRIÇÃO [Biblioteca da função]
AND	• AND lógico [IEC]
NOT	• Inversão de bits [IEC]
OR	• OR lógico [IEC]
XOR	• XOR lógico [IEC]

Movimentação de dados

NOME	• DESCRIÇÃO [Biblioteca da função]
BITW	• Movimentação BIT >>> WORD [ATOS]
BMOVX	• Movimentação indexada na origem e destino [ATOS]
MOVX	• Movimentação indexada no destino [ATOS]
ROL	• Rotação à esquerda [IEC]
ROR	• Rotação à direita [IEC]
SCROLL	• Deslocamento de bloco de dados [ATOS]
SFR	• Deslocamento [ATOS]
SHL	• Deslocamento à esquerda [IEC]
SHR	• Deslocamento à direita [IEC]
TAB	• Carregamento de constantes [ATOS]
WBIT	• Movimentação WORD >>> BIT [ATOS]
WBITX	• Movimentação indexada WORD >>> BIT [ATOS]
WLDX	• Movimentação indexada na origem [ATOS]

Manipulação de strings

NOME	• DESCRIÇÃO [Biblioteca da função]
CONCAT	• Concatenização [IEC]
DELETE	• Apagar [IEC]
FIND	• Procurar [IEC]
INSERT	• Inserir [IEC]
LEFT	• Copia **n** caracteres da string a partir da esquerda [IEC]
LEN	• Comprimento [IEC]
MID	• Cópia [IEC]

REPLACE • Substituir [IEC]

RIGHT • Copia **n** caracteres da string a partir da esquerda [IEC]

Seleção

NOME • **DESCRIÇÃO** [Biblioteca da função]

LIMIT • Limitador de mínimo e máximo [IEC]

MAX • Seleciona maior valor [IEC]

MIN • Seleciona menor valor [IEC]

MUX • Multiplexador extensível [IEC]

SEL • Seleção binária [IEC]

Temporização e contagem

NOME **DESCRIÇÃO** [Biblioteca da função]

CTD • Contador decrescente [IEC]

CTU • Contador crescente [IEC]

CTUD • Contador crescente e decrescente [IEC]

TMR_1MS • Temporizador: *1ms OFF Delay* [ATOS]

TOF • Temporizador: *OFF Delay* [IEC]

TON • Temporizador: *ON Delay* [IEC]

TON_100MS • Temporizador: *100ms ON Delay* [ATOS]

TON_1S • Temporizador: *1s ON Delay* [ATOS]

TP • Temporizador: *Pulse Mode* [IEC]

Trigonométricas

NOME **DESCRIÇÃO** [Biblioteca da função]

ACOS • Arco Cosseno [IEC]

ASIN • Arco Seno [IEC]

ATAN • Arco Tangente [IEC]

COS • Cosseno [IEC]

SIN • Seno [IEC]

TAN • Tangente [IEC]

Especiais

NOME	DESCRIÇÃO [Biblioteca da função]
ADJUST	• Ajuste de escala para entrada analógica [ATOS]
ANGLE COUNTER	• Contador Modo Ângulo [ATOS]
ANGLE SIMULATER	• Simulador de Ângulo [ATOS]
CEP	• Controle Estatístico de Processo [ATOS]
CPU COUNTER	• Contador CPU [ATOS]
EVENT_LOG	• Log de eventos [ATOS]
FILT	• Filtro [ATOS]
HIGH SPEED COUNTER	• Contador Rápido da CPU [ATOS]
PID	• Algoritmo PID padrão ISA [ATOS]
RECIPE	• Arquivo de receitas [ATOS]
RPRINT	• Leitura de canal serial [ATOS]
SCALE	• Ajuste de escala [ATOS]
SCALE_2G	• Ajuste de escala de 2º Grau [ATOS]
SCROLL	• Deslocamento de bloco de dados [ATOS]
SPRINT	• Escrita em canal serial [ATOS]
STEPMOTOR_FASE	• Escrita em canal serial [ATOS]
STEPMOTOR_PULSE	• Escrita em canal serial [ATOS]

CAPÍTULO 2
INSTRUÇÕES DE LÓGICAS

Instruções de lógica entre contatos

• Contato Normalmente Aberto (NA)

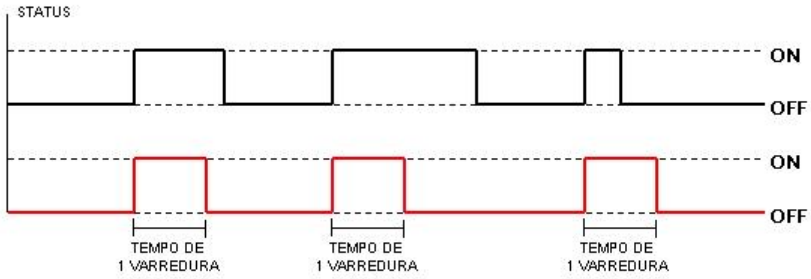
O estado da ligação à esquerda é copiado para a ligação à direita, somente se e o estado da variável booleana associada for ON, caso contrário, o estado da ligação à direita será OFF.

• Contato Normalmente Fechado (NF)

O estado da ligação à esquerda é copiado para a ligação à direita se o estado da variável booleana associada for OFF, caso contrário, o estado da ligação à direita será ON.

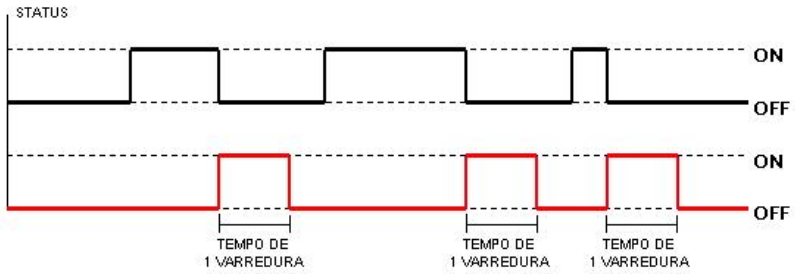
• Contato de Transição Positiva

O estado da ligação à esquerda é copiado para a ligação à direita durante uma varredura se o estado da variável booleana associada for de OFF para ON, caso contrário, o estado da ligação à direita será OFF.
Graficamente temos:



• Contato de Transição Negativa

O estado da ligação à esquerda é copiado para a ligação à direita durante uma varredura se o estado da variável booleana associada for de ON para OFF, caso contrário, o estado da ligação à direita será OFF.
Graficamente temos:



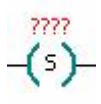
• Saída

O estado da ligação à esquerda é copiado para a ligação à direita e para a variável booleana associada.

• Saída Invertida

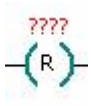
O estado da ligação à esquerda é copiado para a ligação à direita e o inverso da ligação à esquerda para a variável booleana associada.

• Set Saída



O estado da variável booleana associada será setado como ON se a ligação à esquerda for ON e permanecerá assim até o mesmo ser resetado.

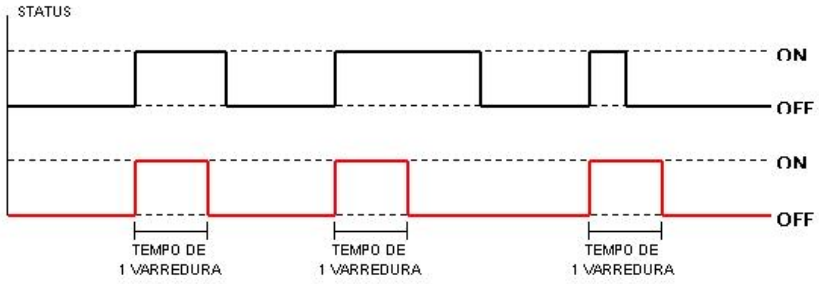
• Reset Saída



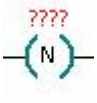
O estado da variável booleana associada será resetado como OFF se a ligação à esquerda for ON e permanecerá assim até o mesmo ser setado.

• Saída de Transição Positiva

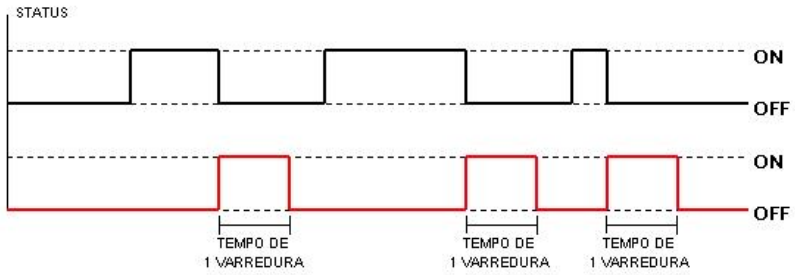
O estado da variável booleana associada será resetado como OFF se a ligação à esquerda for ON e permanecerá assim até o mesmo ser setado.
Graficamente temos:



• Saída de Transição Negativa



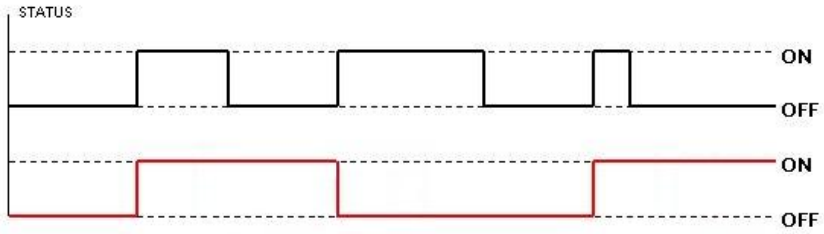
O estado da variável booleana associada será resetado como OFF se a ligação à esquerda for ON e permanecerá assim até o mesmo ser setado.
Graficamente temos:



• Saída SET RESET

O estado da variável booleana associada a saída será setado como ON ou OFF a cada transição positiva (borda de subida) da ligação a esquerda, permanecendo assim até a próxima transição positiva da ligação.

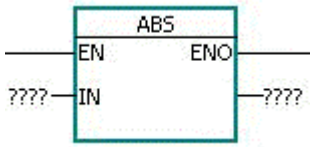
Graficamente temos:



CAPÍTULO 3
INTRUÇÕES IEC61131-3

Aritméticas

ABS - Módulo



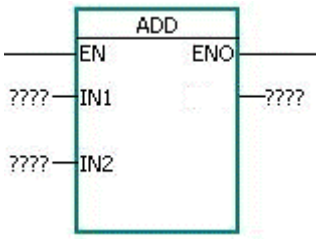
A instrução será executada sempre que EN for habilitada.

$$OUT = |IN|$$

Se não houver erro na execução da instrução, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	INT, DINT, REAL e CONSTANTE	Variável de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, DINT e REAL	Resultado (mesmo tipo de dado de IN).
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.

ADD - Adição



A instrução será executada sempre que a entrada EN estiver habilitada.

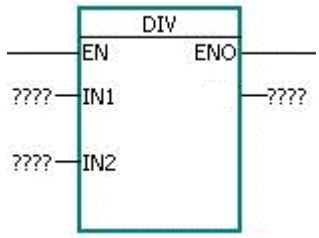
$$OUT = IN1 + IN2$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN1, IN2	INT, DINT, UINT,UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e CONSTANTE	Variáveis de entrada (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, DINT, UINT,UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e CONSTANTE	Resultado (mesmo tipo de dado de IN1 e IN2).
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro de variável.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

DIV - Divisão



A instrução será executada sempre que a entrada EN estiver habilitada.

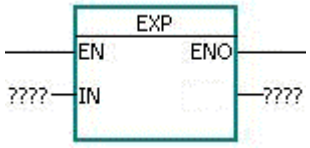
$$OUT = \frac{IN1}{IN2}$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN1, IN2	INT, DINT, UINT, UDINT, WORD, DWORD, REAL e CONSTANTE	Variáveis de entrada (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, DINT, UINT, UDINT, WORD, DWORD, REAL e CONSTANTE	Resultado (mesmo tipo de dado das entradas).
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver divisão por ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

EXP - Exponencial



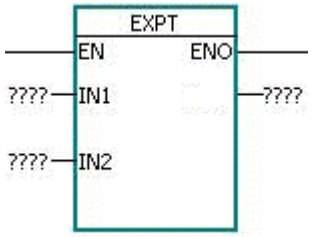
A instrução será executada sempre que a entrada EN estiver habilitada.

$$OUT = e^{IN}$$

Se não houver erro na execução da instrução, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL	Variável de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.

EXPT - Potência X^Y



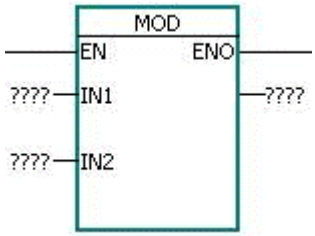
A instrução será executada sempre que a entrada EN estiver habilitada.

$$OUT = (IN1)^{IN2}$$

Se não houver erro na execução da instrução, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN1, IN2	REAL e CONSTANTE	Variáveis de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.

MOD – Módulo de uma divisão

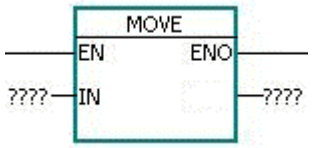


Quando a entrada EN for habilitada, o valor de IN1 é dividido por IN2 (IN1/IN2) e o resto da divisão é colocado na **saída da função**. Se não houver estouro de variável e divisão por zero, a saída ENO será setada, caso contrário será resetada.

$$OUT = MOD\left(\frac{IN1}{IN2}\right)$$

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN1, IN2	INT, DINT, UINT,UDINT, WORD, DWORD, e CONSTANTE	Variáveis de entrada (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Função	INT, DINT, UINT,UDINT, WORD e DWORD	Resultado (mesmo tipo de dado das entradas).
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou divisão por ZERO.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

MOVE - Movimentação de dados



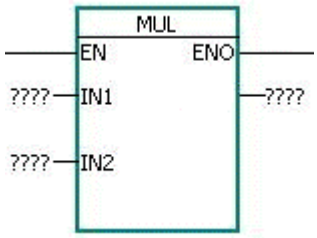
Quando a entrada EN for habilitada, a instrução será executada da seguinte forma:

$$OUT = IN$$

O valor de IN é movido para a **saída da função**.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN	INT, DINT, UINT,UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e CONSTANTE	Variável de origem do dado.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Função	INT, DINT, UINT,UDINT, WORD, DWORD, REAL, TIME, DATE e TOD	Variável de destino do dado (mesmo tipo de dado da entrada).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

MUL - Multiplicação



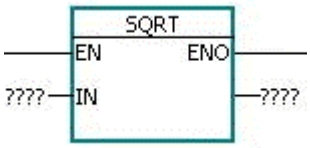
Quando a entrada EN for habilitada, o valor de IN1 é multiplicado por IN2 (IN1*IN2) e o resultado é colocado na **saída da função**. Se não houver estouro de variável, a saída ENO será setada, caso contrário será resetada.

$$OUT = IN1 \cdot IN2$$

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN1, IN2	INT, DINT, UINT,UDINT, WORD, DWORD, REAL e CONSTANTE	Variáveis de entrada (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, DINT, UINT,UDINT, WORD, DWORD e REAL.	Resultado (mesmo tipo de dado das entradas).
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro de variável.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

SQRT - Raiz Quadrada



A instrução será executada sempre que a entrada EN estiver habilitada.

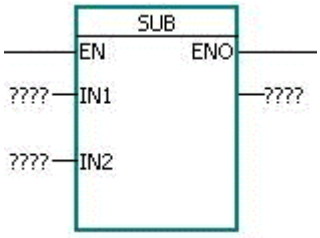
$$OUT = \sqrt{IN}$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

Dependendo do número de caracteres e precisão do campo Edit/View configurado na IHM, pode haver arredondamento do valor.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL	Variáveis de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída, se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado ou se o valor da entrada IN for NEGATIVO.
	N	**Sinal**	Será ligado se a variável de entrada for NEGATIVA.
	Z	**Zero**	Será ligado se o resultado for ZERO.

SUB - Subtração



A instrução será executada sempre que a entrada EN estiver habilitada.

$$OUT = IN1 - IN2$$

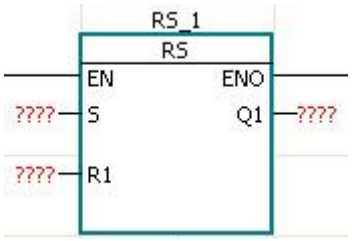
Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN1, IN2	INT, DINT, UINT,UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e CONSTANTE	Variáveis de entrada (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, DINT, UINT,UDINT, WORD, DWORD, REAL, TIME, DATE e TOD	Resultado (mesmo tipo de dado das entradas).
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro de variável.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

Biestáveis

RS – Biestável com Reset dominante



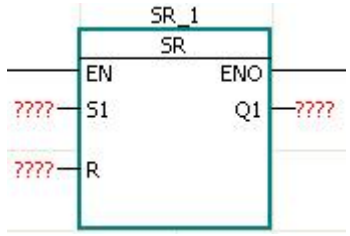
A instrução será executada sempre que a entrada EN estiver habilitada.

Durante a execução é verificado a alteração do estado de R1 e S sendo que, R1 é dominante sobre S. Quando R1=True a saída Q1 é desacionada independente do estado de S (True ou False).

Para acionar a saída Q1: S=True e R1=False.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	S	BOOL	Entrada Set.
	R1	BOOL	Reset dominante.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Q1	BOOL	Saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

SR – Biestável com Set dominante



A instrução será executada sempre que a entrada EN estiver habilitada.

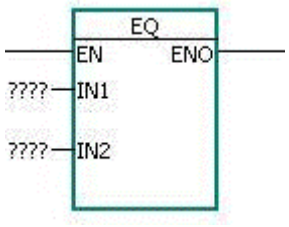
Durante a execução é verificado a alteração do estado de S1 e R sendo que, S1 é dominante sobre R. Quando S1=True a saída Q1 é acionada independente do estado de R (True ou False).

Para desacionar a saída Q1: S1=False e R=True.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	S1	BOOL	Entrada Set dominante.
	R	BOOL	Reset.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Q1	BOOL	Saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

Comparação

EQ - Igual



Quando a entrada EN for habilitada, é feita a comparação das entradas IN1 e IN2, se o resultado for **IGUAL**, a saída ENO será ligada, caso contrário será desligada.

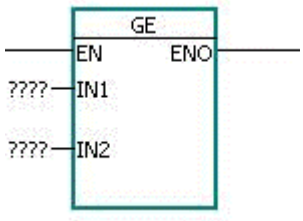
$$ENO = (IN1 = IN2)$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN1, IN2	INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, DATE, TOD e CONSTANTE.	Variáveis de comparação (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Resultado da comparação
	FLAG	NOME	DESCRIÇÃO
	-----	-----	Nenhum *flag* é afetado

GE - Maior ou igual que



Quando a entrada EN for habilitada, é feita a comparação das entradas IN1 e IN2, se a entrada IN1 for **MAIOR OU IGUAL QUE** IN2, a saída ENO será ligada, caso contrário será desligada.

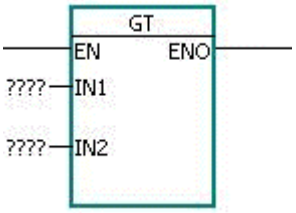
$$ENO = (IN1 \geqslant IN2)$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN1, IN2	INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, DATE, TOD e CONSTANTE.	Variáveis de comparação (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Resultado da comparação
	FLAG	NOME	DESCRIÇÃO
	-----	-----	Nenhum *flag* é afetado

GT - Maior que



Quando a entrada EN for habilitada, é feita a comparação das entradas IN1 e IN2, se a entrada IN1 for **MAIOR QUE** IN2, a saída ENO será ligada, caso contrário será desligada.

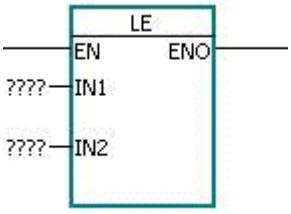
$$ENO = (IN1 > IN2)$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN1, IN2	INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, DATE, TOD e CONSTANTE.	Variáveis de comparação (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Resultado da comparação
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

LE - Menor ou igual que



Quando a entrada EN for habilitada, é feita a comparação das entradas IN1 e IN2, se a entrada IN1 for **MENOR OU IGUAL QUE** IN2, a saída ENO será ligada, caso contrário será desligada.

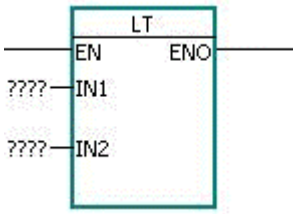
$$ENO = (IN1 \leqslant IN2)$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN1, IN2	INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, DATE, TOD e CONSTANTE.	Variáveis de comparação (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Resultado da comparação
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

LT - Menor que



Quando a entrada EN for habilitada, é feita a comparação das entradas IN1 e IN2, se a entrada IN1 for **MENOR QUE** IN2, a saída ENO será ligada, caso contrário será desligada.

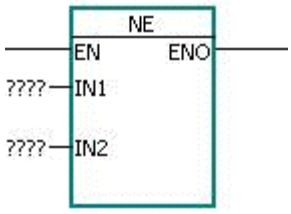
$$ENO = (IN1 < IN2)$$

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN1, IN2	INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, DATE, TOD e CONSTANTE.	Variáveis de comparação (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Resultado da comparação
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

NE - Diferente



Quando a entrada EN for habilitada, é feita a comparação das entradas IN1 e IN2, se o resultado for **DIFERENTE**, a saída ENO será ligada, caso contrário será desligada.

$$ENO = (IN1 \neq IN2)$$

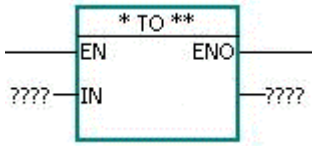
Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

OBS: As entradas IN1 e IN2 da instrução devem ser sempre do mesmo tipo de dado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN1, IN2	INT, UINT, WORD, DINT, UDINT, DWORD, REAL, TIME, DATE, TOD e CONSTANTE.	Variáveis de comparação (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Resultado da comparação
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

Conversão

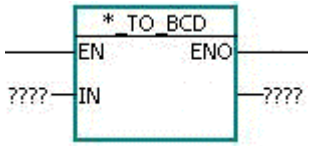
*_TO_**



Quando a entrada EN for habilitada, é feita a conversão do conteúdo de IN e colocado o valor na **saída da função**. Se não houver erro de conversão, a saída ENO ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	INT, DINT, UINT, UDINT, WORD e DWORD, REAL, TIME, DATE e TOD.	Valor a ser convertido.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, DINT, UINT, UDINT, WORD, DWORD, REAL, TIME, DATE e TOD.	Valor convertido.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro de variável.
	N	**Sinal**	Se o valor de IN for negativo, será colocado em OUT o módulo do valor convertido para os tipos sem sinal e este flag será ligado.

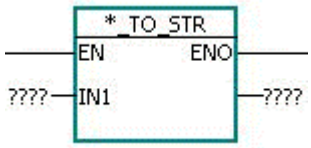
*_TO_BCD



Quando a entrada EN for habilitada, é feita a conversão do conteúdo de IN e colocado o valor na **saída da função**. Se não houver erro de conversão, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	UINT, UDINT, INT e DINT	Valor a ser convertido p/ BCD.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	BCD (WORD ou DWORD)	Valor convertido em BCD (deve ser armazenado em uma variável do tipo de dado WORD ou DWORD).
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se o valor de IN for maior que 9999 (WORD) ou 99999999 (DWORD).
	N	**Sinal**	Se o valor de IN for negativo, será colocado em OUT o módulo do valor convertido e este flag será ligado.

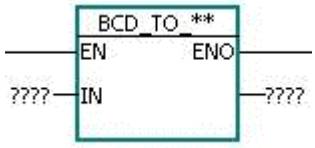
*_TO_STR – Converte para STRING



Quando a entrada de EN for habilitada, será feita a conversão do tipo de dado da entrada IN para a **saída da função**. Se não houver erro de conversão, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	INT, DINT, UINT, UDINT, WORD, DWORD e REAL.	Valor a ser convertido em string.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	STRING	Variável para armazenamento da string convertida.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

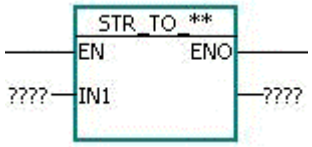
BCD_TO_**



Quando a entrada EN for habilitada, é feita a conversão do conteúdo de IN e colocado o valor na **saída da função**. Se não houver erro de conversão, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN	BCD (WORD ou DWORD)	Valor em BCD a ser convertido (deve estar armazenado em uma variável de tipo de dado WORD ou DWORD)
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Função	UINT, UDINT, INT e DINT	Valor convertido para o tipo de dado escolhido (UINT, UDINT, INT e DINT).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

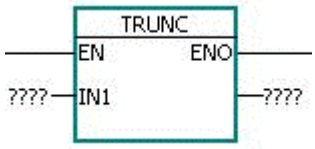
STR_TO_** - Converte STRING para INT/DINT



Quando a entrada de EN for habilitada, será feita a conversão do tipo de dado da entrada IN para a **saída da função**. Se não houver erro de conversão, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	STRING	String a ser convertida em um valor numérico.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT e DINT	String convertida para o tipo de dado escolhido (INT ou DINT).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

TRUNC



Quando a entrada EN for habilitada, é truncada a parte decimal do conteúdo de IN e colocado o valor na **saída da função**. Se não houver erro de conversão, a saída ENO será ligada, caso contrário será desligada.

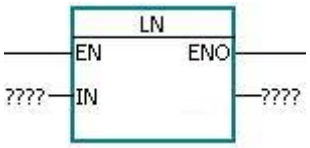
Exemplos de truncamento:

- TRUNC (1,6) **é equivalente a** 1;
- TRUNC (-1,6) **é equivalente a** -1;
- TRUNC (1,4) **é equivalente a 1**;
- TRUNC (-1,4) **é equivalente a** -1.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL	Valor a ser arredondado.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, DINT, UINT e UDINT.	Valor arredondado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro de variável.
	N	**Sinal**	Se o valor de IN for negativo, será colocado em OUT o módulo do valor convertido para os tipos sem sinal e este flag será ligado.

Logarítmicas

LN - Logaritmo natural



A instrução será executada sempre que a entrada EN estiver habilitada.

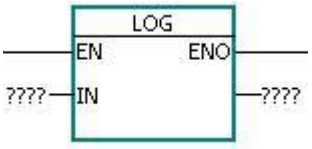
$$OUT = LN\ (IN)$$

O resultado estará sempre entre os valores:
-87,33654 ≤ **OUT** ≤ 88,72284.

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL	Variável de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

LOG - Logaritmo de base 10



A instrução será executada sempre que a entrada EN estiver habilitada.

$$OUT = LOG\ (IN)$$

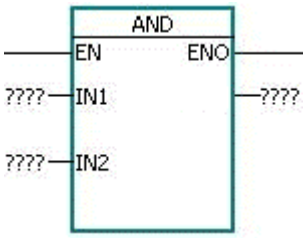
O resultado estará sempre entre os valores:
-37,92978 ≤ **OUT** ≤ 38,53184.

Após a execução da instrução, se não houver estouro de variável, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL	Variável de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

Lógica entre registros

AND

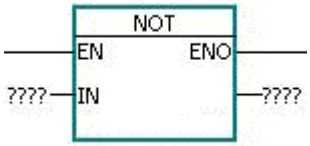


A instrução será executada sempre que a entrada EN estiver habilitada.

$$OUT = IN1 \text{ AND } IN2$$

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN1, IN2	WORD, DWORD e CONSTANTE	Variáveis de entrada (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	WORD e DWORD	Resultado (mesmo tipo de dado das entradas).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

NOT



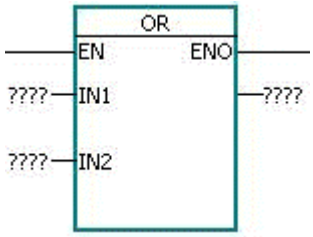
A instrução será executada sempre que a entrada EN estiver habilitada.

$$OUT = NOT(IN)$$

É feita uma inversão dos bits de IN e o resultado, colocado na **saída da função**.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	WORD, DWORD e CONSTANTE	Variáveis de entrada (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	OUT	WORD e DWORD	Resultado (mesmo tipo de dado das entradas).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

OR

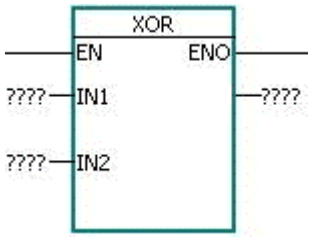


A instrução será executada sempre que a entrada EN estiver habilitada.

OUT = IN1 OR IN2

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN1, IN2	WORD, DWORD e CONSTANTE	Variáveis de entrada (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Função	WORD e DWORD	Resultado (mesmo tipo de dado das entradas).
	FLAG	NOME	DESCRIÇÃO
	-----	-----	Nenhum *flag* é afetado

XOR



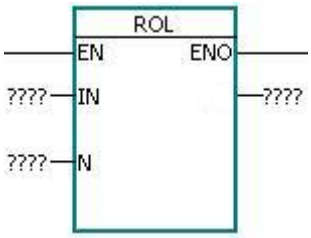
A instrução será executada sempre que a entrada EN estiver habilitada.

OUT = IN1 XOR IN2

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN1, IN2	WORD, DWORD e CONSTANTE	Variáveis de entrada (ambas as entradas devem ser do mesmo tipo de dado).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	WORD e DWORD	Resultado (mesmo tipo de dado das entradas).
	FLAG	NOME	DESCRIÇÃO
	-----	-----	Nenhum *flag* é afetado.

Movimentação de dados

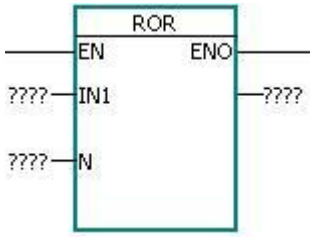
ROL - Rotação à esquerda



Quando a entrada EN for habilitada, o valor de IN será rotacionado à esquerda N vezes de forma circular.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN	WORD, DWORD e CONSTANTE	Variável com conteúdo a ser rotacionado.
	N	UINT e CONSTANTE	Número de rotações à esquerda dos bits da variável definida em IN.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Função	WORD e DWORD	Valor rotacionado de N vezes (mesmo tipo de dado da entrada IN).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

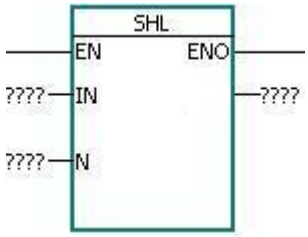
ROR - Rotação à direita



Quando a entrada EN for habilitada, o valor de IN será rotacionado à direita N vezes de forma circular.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN	WORD, DWORD e CONSTANTE	Variável com conteúdo a ser rotacionado.
	N	UINT e CONSTANTE	Número de rotações à direita dos bits da variável definida em IN.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Função	WORD e DWORD	Valor rotacionado de N vezes (mesmo tipo de dado da entrada IN).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

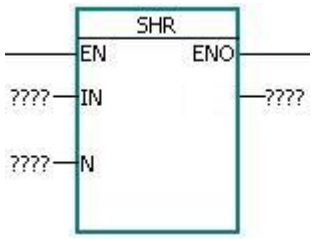
SHL - Deslocamento à esquerda



Quando a entrada EN for habilitada, o valor de IN será deslocado a esquerda N vezes e inserido ZEROS à direita.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN	WORD, DWORD e CONSTANTE	Variável com conteúdo a ser deslocado.
	N	UINT e CONSTANTE	Número de deslocamentos à esquerda dos bits da variável definida em IN.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Função	WORD e DWORD	Valor deslocado de N vezes (mesmo tipo de dado da entrada IN).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

SHR - Deslocamento à direita

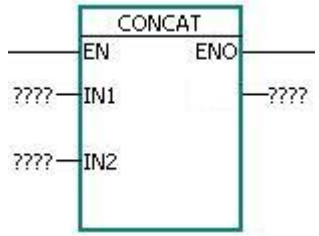


Quando a entrada EN for habilitada, o valor de IN1 será deslocado a direita N vezes e inserido ZEROS à esquerda.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	WORD, DWORD e CONSTANTE	Variável com conteúdo a ser deslocado.
	N	UINT e CONSTANTE	Número de deslocamentos à direita dos bits da variável definida em IN.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	WORD e DWORD	Valor deslocado de N vezes (mesmo tipo de dado da entrada IN).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Manipulação de Strings

CONCAT - Concatenação



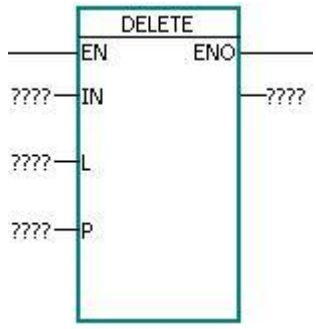
Quando a entrada de EN for habilitada, a **saída da função** conterá o conterá o conteúdo da entrada IN1 mais o conteúdo da entrada IN2, e o seu tamanho atual será a soma do tamanho das duas entradas.

Se o comprimento total das duas entradas for maior que o da variável de saída, será truncada parte da entrada IN2 e será ligado o *flag* de *Overflow*.

A saída ENO será uma cópia da entrada EN se não houver nenhum erro.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN1, IN2	STRING	Strings a serem concatenadas.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	STRING	String com o conteúdo de IN1+IN2.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se o comprimento total das duas entradas for maior que o da variável de saída.

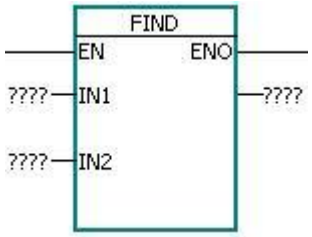
DELETE - Apagar



Quando a entrada de EN for habilitada, será colocado na **saída da função** o conteúdo de IN menos L caracteres a partir da posição P. O seu tamanho atual será o seu tamanho menos o número de caracteres definido na entrada L. A saída ENO será uma cópia da entrada EN.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	STRING	String de origem.
	L	UINT	Quantidade de caracteres para apagar.
	P	UINT	Posição na String IN.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	STRING	String de destino.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

FIND - Procurar

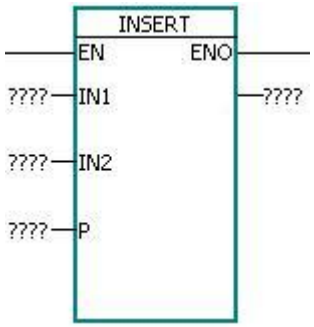


Quando a entrada de EN for habilitada, será colocado na **saída da função**, a posição onde foi encontrada a ocorrência da entrada IN2 dentro da entrada IN1, se não for encontrada esta ocorrência, será retornado "0". A saída ENO será uma cópia da entrada EN.

Se o conteúdo de IN2 for vazio, a saída ENO será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN1, IN2	STRING	String de origem e de procura.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	UINT	Posição onde IN2 foi encontrado em IN1.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

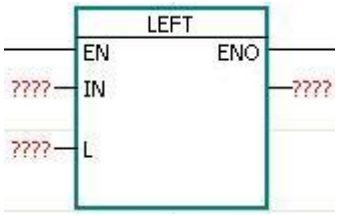
INSERT - Inserir



Quando a entrada de EN for habilitada, será colocado na **saída da função** o conteúdo de IN1 e será inserido na posição P o conteúdo da entrada IN2. O seu tamanho atual será a soma do tamanho das duas entradas. Se o comprimento total das duas entradas for maior que o da variável de saída, será truncada o final da variável e será ligado o ***flag* de *Overflow***. A saída ENO será uma cópia da entrada EN.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	STRING	Strings de origem.
	P	UINT	Posição para inserir IN2 em IN1.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	STRING	String de destino para o resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	Overflow	**OV**	Será ligado quando o valor da entrada **P** for maior que o número de caracteres da string.

LEFT – Copia n caracteres da string a partir da esquerda

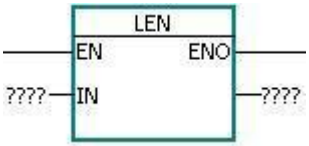


A instrução será executada sempre que a entrada EN estiver habilitada.

Copia **n** posições da entrada **IN** a partir da primeira posição da string até a última posição. O resultado dessa seleção é copiado para a **saída da função**.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	STRING	Entrada.
	L	UINT	Número de posições.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	STRING	Saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	--------	--------	Nenhum *flag* é afetado.

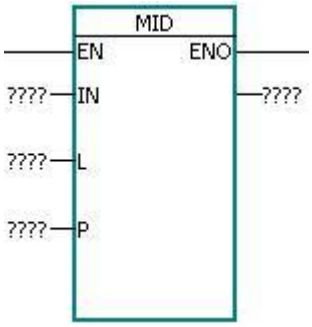
LEN - Comprimento



Quando a entrada de EN for habilitada, a **saída da função** conterá o comprimento atual da STRING colocada na entrada IN. A saída ENO será uma cópia da entrada EN.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	STRING	String de origem.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	UINT	Tamanho da String declarada em IN.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

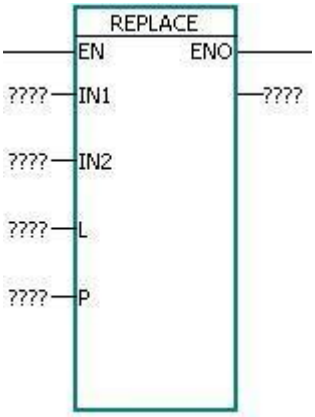
MID - Copiar



Quando a entrada de EN for habilitada, será colocado na **saída da função**, L caracteres de IN a partir da posição P. A saída ENO será uma cópia da entrada EN.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	STRING	String de origem.
	L	UINT	Quantidade de caracteres.
	P	UINT	Posição na string IN.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	STRING	Valor copiado da String em IN.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

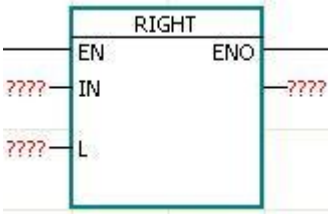
REPLACE - Substituir



Quando a entrada de EN for habilitada, será colocado na **saída da função** uma cópia de IN1, porém serão substituídos L caracteres a partir da posição P pelo conteúdo da entrada IN2. A saída ENO será uma cópia da entrada EN.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	STRING	Strings de origem.
	L	UINT	Quantidade de caracteres.
	P	UINT	Posição de IN1.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	STRING	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

RIGHT – Copia n caracteres da string a partir da direita



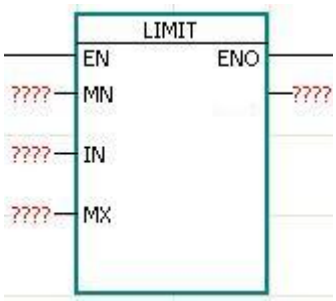
A instrução será executada sempre que a entrada EN estiver habilitada.

Copia **n** posições da entrada **IN** a partir da última posição da string até a primeira posição. O resultado dessa seleção é copiado para a **saída da função**.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	STRING	Entrada.
	L	UINT	Número de posições.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	STRING	Saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	--------	--------	Nenhum *flag* é afetado.

Seleção

LIMIT – Limitador de mínimo e máximo



A instrução será executada sempre que a entrada EN estiver habilitada.

OUT = Min[Máx(IN, MN), MX]

Verifica se a entrada IN está entre o mínimo e o máximo determinado pelas entradas MN e MX. Se IN estiver entre os valores de MN e MX a entrada será copiada para a **saída da função**.

Se **MN < IN < MX** então **OUT = IN**

Para outras configurações têm-se:

- Se IN for menor que o definido por MN, o valor de MN será copiado para a **saída da função**;

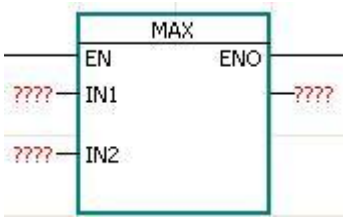
  Se **IN < MN** então **OUT = MN**

- Se IN for maior que o definido por MX, o valor de MX será copiado para a **saída da função**.

  Se **IN > MX** então **OUT = MX**

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	MN	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e DATE_AND_TIME	Valor mínimo.
	In		Entrada.
	MX		Valor máximo.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e DAT	Saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	--------	--------	Nenhum *flag* é afetado.

MAX – Seleciona maior valor



A instrução será executada sempre que a entrada EN estiver habilitada.

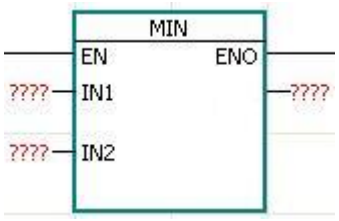
OUT = MAX (IN1, IN2, IN#)

Determina o maior valor entra as entradas IN (mín. 2 e máx 32) e copia para a **saída da função**.

Para configurar um número maior de parâmetros (Entradas), selecione o bloco funcional com o cursor do mouse (área de programação do Ladder), clique com o botão direito e marque a opção ***Quantidade de Parâmetros***.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	In1	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e DATE_AND_TIME	Entrada 1.
	In2		Entrada 2.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e DATE_AND_TIME	Saída copiada depois da comparação entre o maior valor de In1 e In2.
	FLAG	**NOME**	**DESCRIÇÃO**
	--------	--------	Nenhum *flag* é afetado.

MIN – Seleciona menor valor



A instrução será executada sempre que a entrada EN estiver habilitada.

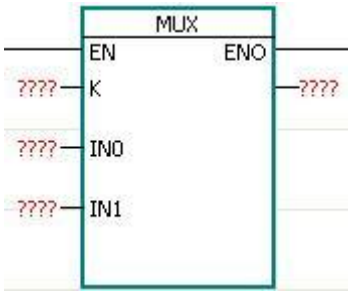
OUT = MIN (IN1, IN2, IN#)

Determina o menor valor entra as entradas IN (mín. 2 máx 32) e copia para a **saída da função**.

Para configurar um número maior de parâmetros (Entradas), selecione o bloco funcional com o cursor do mouse (área de programação do Ladder), clique com o botão direito e marque a opção ***Quantidade de Parâmetros.***

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	In1	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e DAT	Entrada 1.
	In2		Entrada 2.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e DATE_AND_TIME	Saída copiada depois da comparação entre o menor valor de In1 e In2.
	FLAG	**NOME**	**DESCRIÇÃO**
	--------	--------	Nenhum *flag* é afetado.

MUX – Multiplexador extensível



A instrução será executada sempre que a entrada EN estiver habilitada.

OUT = IN [K]

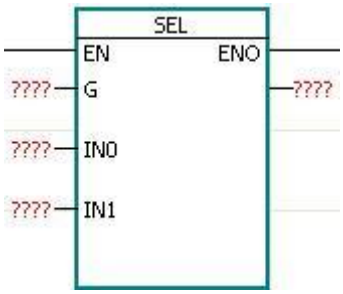
Seleciona dentre as **N** entradas IN (mín. 2 e máx 31) uma a ser copiada para a **saída da função**. A entrada selecionada é indicada por **K**.

Para configurar um número maior de parâmetros (Entradas), selecione o bloco funcional com o cursor do mouse (área de programação do Ladder), clique com o botão direito e marque a opção ***Quantidade de Parâmetros***.

Para selecionar uma entrada, o valor de K deve ser correspondente à entrada, ou seja, para IN0 setar K=0, para IN5 setar K=5, para IN30 setar K=30 e etc.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	K	UINT	Seleciona entrada a ser copiada.
	In1	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e DAT	Entrada 1.
	In2		Entrada 2.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD e DATE_AND_TIME	Saída copiada depois de setado número da entrada em K.
	FLAG	**NOME**	**DESCRIÇÃO**
	--------	--------	Nenhum *flag* é afetado.

SEL – Seleção binária



A instrução será executada sempre que a entrada EN estiver habilitada.

Seleciona um valor entre duas entradas (IN0 e IN1). Essa seleção é feita pela entrada **G**. Caso a entrada **G** esteja desligada a **saída da função** recebe o valor de IN0, caso a entrada **G** esteja ligada a **saída da função** recebe o valor de IN1.

OUT = IN0 se **G = False**

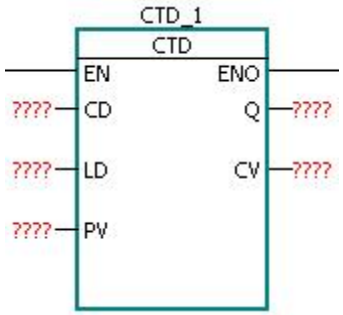
OUT = IN1 se **G = True**

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	G	BOOL	Seleciona entrada.
	IN0	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD, DATE_AND_TIME e STRING	Entrada.
	IN1		
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	INT, UINT, DINT, UDINT, WORD, DWORD, REAL, TIME, DATE, TOD, DATE_AND_TIME e STRING	Saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	--------	--------	Nenhum *flag* é afetado.

Temporização e Contagem

CTD - Contador Decrescente

O Bloco Funcional CTD pode ser utilizado de dois modos distintos: Com e sem EN/ENO. Cada modo possui uma característica diferente de trabalho e podem ser utilizados individualmente para cada CTD inserido no programa.



Modo: Com EN/ENO

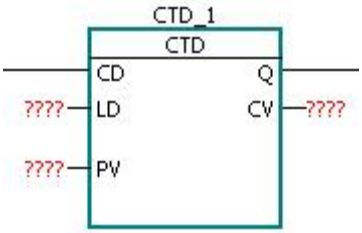
Com a entrada **EN** habilitada, a cada pulso em **CD** a saída **CV** será decrementada.

Quando o valor de contagem atingir o valor **"0"** zero a saída **Q** será ligada, caso contrário permanecerá desligada. A contagem permanece até a saída **CV** atingir o valor CV=0, o contador permanecerá parado mesmo com pulsos na entrada CD até que seja carregado (**LD**) um novo valor em CV.

Quando a entrada load (**LD**) for habilitada, será carregado na saída de contagem (**CV**) o valor do preset (**PV**). Tanto PV quanto CV devem ser do mesmo tipo de dado.

Pulsos na entrada CD ou acionamento de LD somente serão interpretados pela instrução se a entrada **EN** estiver habilitada.

Durante a contagem é possível a mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua decrementando a saída CV até CV=0. Caso a entrada LD seja acionada durante a contagem, será carregado automaticamente o novo valor do preset (PV) para a saída CV e o bloco funcional continua a contagem a partir do novo valor de CV.



Modo: Sem EN/ENO

A cada pulso em **CD** a saída **CV** será decrementada.

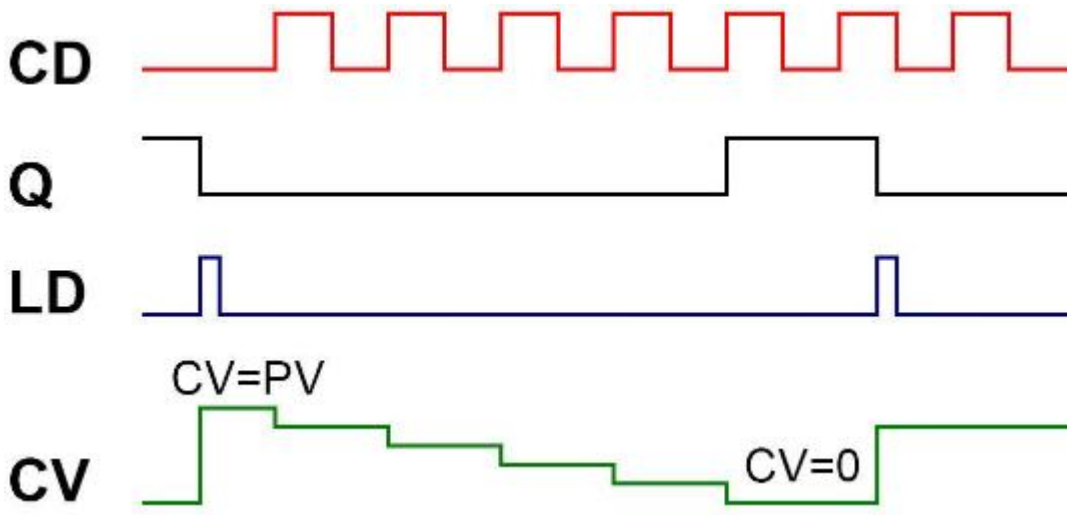
Quando o valor de contagem atingir o valor **"0"** zero a saída **Q** será ligada, caso contrário permanecerá desligada. A contagem permanece até a saída **CV** atingir o valor CV=0, o contador permanecerá parado mesmo com pulsos na entrada CD até que seja carregado (LD) um novo valor em CV.

Quando a entrada load (**LD**) for habilitada, será carregado na saída de contagem (**CV**) o valor do preset (**PV**). Tanto PV quanto CV devem ser do mesmo tipo de dado.

Durante a contagem é possível a mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua decrementando a saída CV até CV=0. Caso a entrada LD seja acionada durante a contagem, será carregado automaticamente o novo valor do preset (PV) para a saída CV e o bloco funcional continua a contagem a partir do novo valor de CV.

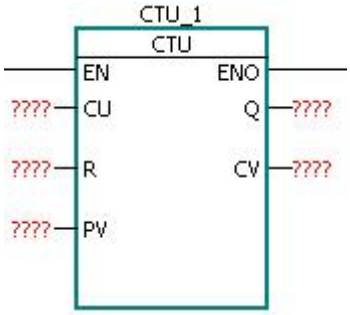
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Entrada **opcional**. Existente somente no modo com EN/ENO.
	CD	BOOL	Sinal de contagem (pulso).
	LD	BOOL	Carrega Preset (PV) em CV (efetivo).
	PV	INT, UINT, DINT e UDINT	Preset do contador.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Saída **opcional**. Existente somente no modo com EN/ENO.
	Q	BOOL	Saída do contador.
	CV	INT, UINT, DINT e UDINT	Efetivo do contador.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Exemplo gráfico de funcionamento:



CTU - Contador Crescente

O Bloco Funcional CTU pode ser utilizado de dois modos distintos: Com e sem EN/ENO. Cada modo possui uma característica diferente de trabalho e podem ser utilizados individualmente para cada CTU inserido no programa.



Modo: Com EN/ENO

Com a entrada **EN** habilitada, a cada pulso em **CU** a saída **CV** será incrementada.

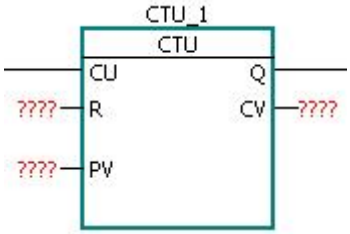
Enquanto o valor da saída **CV** for menor que o valor de preset (**PV**), a saída **Q** permanecerá desligada. Quando a saída **CV** for igual ao valor de preset (**PV**), a saída **Q** será ligada.

Quando a entrada de reset (**R**) for habilitada, o contador será zerado, isto é, CV=0.

A contagem permanece até a saída CV=PV. Atingindo este valor, o contador permanecerá parado mesmo com pulsos na entrada **CU** até que seja resetada (**R**) a saída **CV**.

Pulsos na entrada **CU** ou acionamento de **R** somente serão interpretados pelo bloco funcional se a entrada **EN** estiver habilitada.

Durante a contagem é possível a mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua incrementando a saída CV até CV=PV caso o novo valor de PV seja maior que CV. Sendo o novo valor do preset PV menor que CV, então a contagem é interrompida e a saída Q acionada.



Modo: Sem EN/ENO

A cada pulso em **CU**, a saída **CV** será incrementada.

Enquanto o valor da saída **CV** for menor que o valor de preset (**PV**), a saída **Q** permanecerá desligada. Quando a saída **CV** for igual ao valor de preset (**PV**), a saída **Q** será ligada.

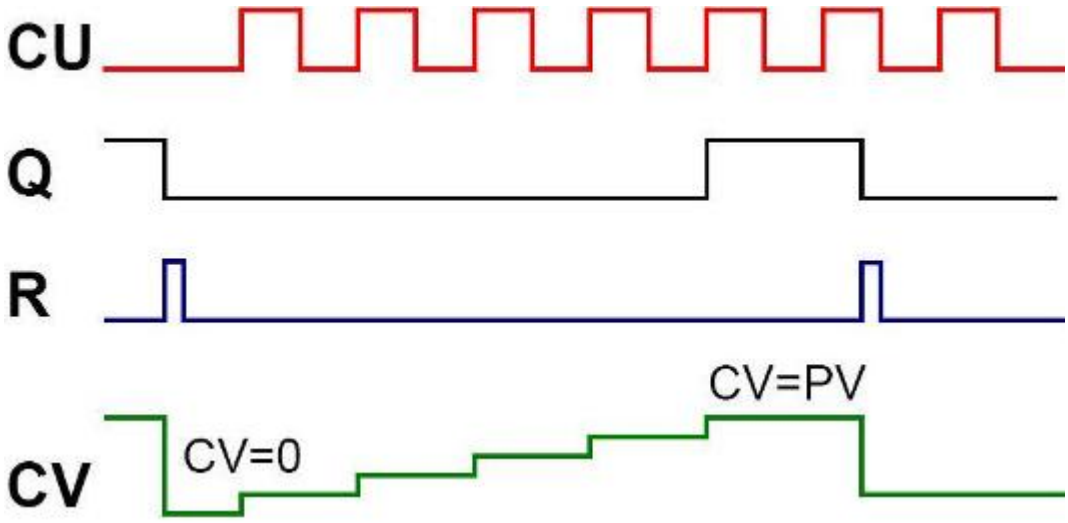
Quando a entrada de reset **R** for habilitada, o contador será zerado, isto é, CV=0.

A contagem permanece até a saída CV=PV. Atingindo este valor, o contador permanecerá parado mesmo com pulsos na entrada **CU** até que seja resetada (**R**) a saída **CV**.

Durante a contagem é possível a mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua incrementando a saída CV até CV=PV caso o novo valor de PV seja maior que CV. Sendo o novo valor do preset PV menor que CV, então a contagem é interrompida e a saída Q acionada.

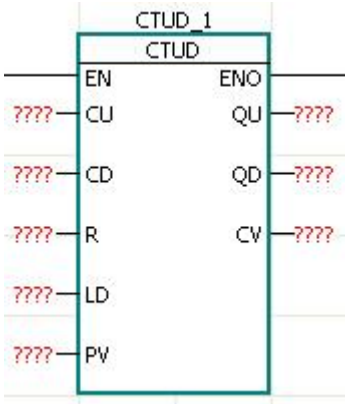
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Entrada **opcional**. Existente somente no modo com EN/ENO.
	CU	BOOL	Sinal de contagem (pulso).
	R	BOOL	Reset de contagem (CV = 0).
	PV	INT, UINT, DINT e UDINT	Preset de contagem.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Saída **opcional**. Existente somente no modo com EN/ENO.
	Q	BOOL	Saída do contador.
	CV	INT, UINT, DINT e UDINT	Efetivo de contagem.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

Exemplo gráfico de funcionamento:



CTUD - Contador Crescente e Decrescente

O Bloco Funcional CTU pode ser utilizado de dois modos distintos: Com e sem EN/ENO. Cada modo possui uma característica diferente de trabalho e podem ser utilizados individualmente para cada CTUD inserido no programa.



Modo: Com EN/ENO

Com a entrada **EN** habilitada, a instrução pode assumir contagem crescente (pulsos em CU) ou decrescente (pulsos em CD).

OBS.: Pulsos aplicados simultaneamente nas entradas CU e CD não serão interpretados pela instrução, fazendo com que a mesma não execute nenhum dos modos de contagem.

Contagem crescente

Com a entrada **EN** habilitada, a cada pulso em **CU** a saída **CV** será incrementada.

Enquanto o valor da saída **CV** for menor que o valor de preset (**PV**), a saída **QU** permanecerá desligada. Quando a saída **CV** for igual ao valor de preset (**PV**), a saída **QU** será ligada.

Quando a entrada de reset (**R**) for habilitada, o contador será zerado, isto é, CV=0.

A contagem permanece até a saída CV=PV. Atingindo este valor, o contador permanecerá parado mesmo com pulsos na entrada **CU** até que seja resetada (**R**) a saída **CV**.

Pulsos na entrada CU ou acionamento de R somente serão interpretados pela instrução se a entrada EN estiver habilitada.

Durante a contagem é possível a mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua incrementando a saída CV até CV=PV caso o novo valor de PV seja maior que CV. Sendo o novo valor do preset PV menor que CV, então a contagem é interrompida e a saída QU acionada.

Contagem decrescente

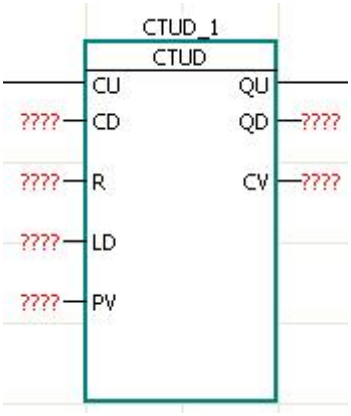
Com a entrada **EN** habilitada, a cada pulso em **CD** a saída **CV** será decrementada.

Quando o valor de contagem atingir o valor **"0"** zero a saída **QD** será ligada, caso contrário permanecerá desligada. A contagem permanece até a saída **CV** atingir o valor CV=0, o contador permanecerá parado mesmo com pulsos na entrada CD até que seja carregado (**LD**) um novo valor em CV.

Quando a entrada load (**LD**) for habilitada, será carregado na saída de contagem (**CV**) o valor do preset (**PV**). Tanto PV quanto CV devem ser do mesmo tipo de dado.

Pulsos na entrada CD ou acionamento de LD somente serão interpretados pela instrução se a entrada **EN** estiver habilitada.

Durante a contagem é possível a mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua decrementando a saída CV até CV=0. Caso a entrada LD seja acionada durante a contagem, será carregado automaticamente o novo valor do preset (PV) para a saída CV e o bloco funcional continua a contagem a partir do novo valor de CV.



Modo: Sem EN/ENO

Sem a entrada **EN** a instrução estará sempre habilitada.

Pulsos para contagem crescente devem ser aplicados na entrada CU e para contagem decrescente aplicados na entrada CD.

OBS.: Pulsos aplicados simultaneamente nas entradas CU e CD não serão interpretados pela instrução, fazendo com que a mesma não execute nenhum dos modos de contagem.

Contagem crescente

A cada pulso em CU, a saída CV será incrementada.

Enquanto o valor da saída **CV** for menor que o valor de preset (**PV**), a saída **Q** permanecerá desligada. Quando a saída **CV** for igual ao valor de preset (**PV**), a saída **QU** será ligada.

Quando a entrada de reset **R** for habilitada, o contador será zerado, isto é, CV=0.

A contagem permanece até a saída CV=PV. Atingindo este valor, o contador permanecerá parado mesmo com pulsos na entrada **CU** até que seja resetada (**R**) a saída **CV**.

Durante a contagem é possível a mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua incrementando a saída CV até CV=PV caso o novo valor de PV seja maior que CV. Sendo o novo valor do preset PV menor que CV, então a contagem é interrompida e a saída QU acionada.

Contagem decrescente

A cada pulso em CD a saída CV será decrementada.

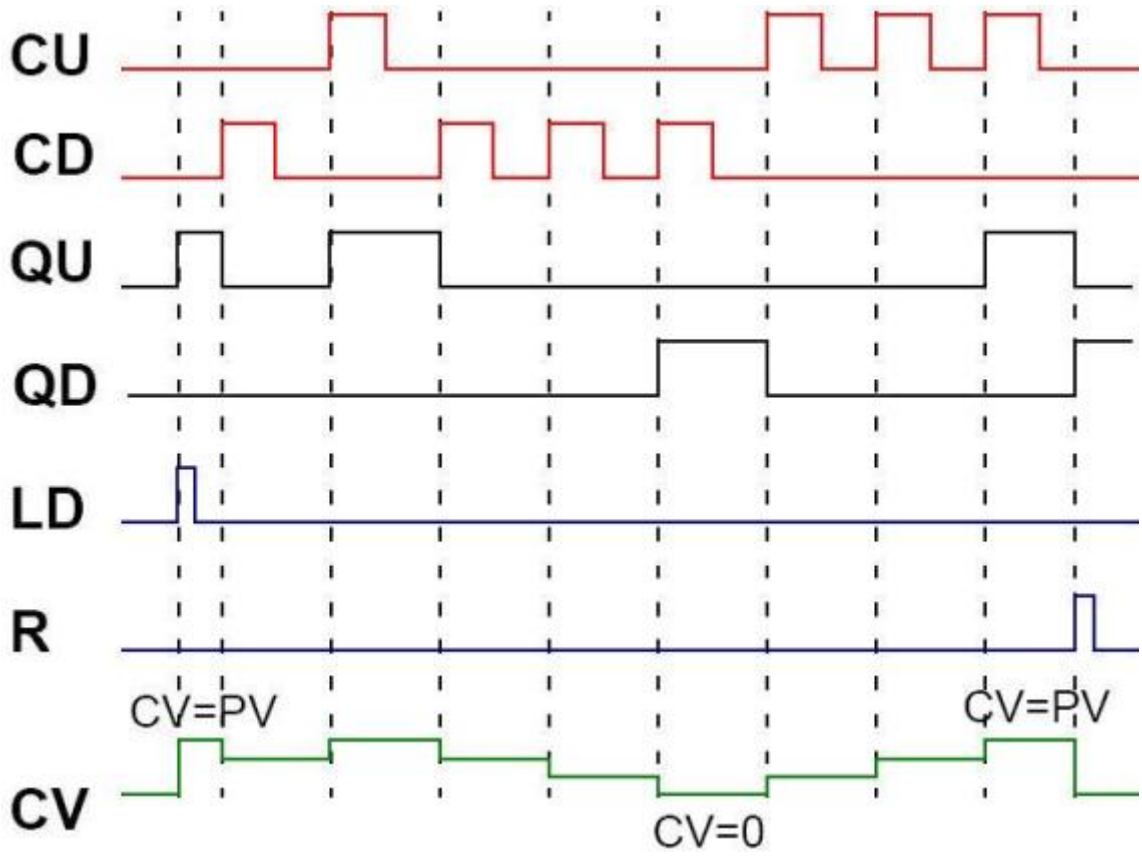
Quando o valor de contagem atingir o valor **"0"** zero a saída **QD** será ligada, caso contrário permanecerá desligada. A contagem permanece até a saída **CV** atingir o valor CV=0, o contador permanecerá parado mesmo com pulsos na entrada CD até que seja carregado (LD) um novo valor em CV.

Quando a entrada load (**LD**) for habilitada, será carregado na saída de contagem (**CV**) o valor do preset (**PV**). Tanto PV quanto CV devem ser do mesmo tipo de dado.

Durante a contagem é possível a mudança de valor do preset (PV). Alterando o valor do preset (PV) durante a contagem, o bloco funcional continua decrementando a saída CV até CV=0. Caso a entrada LD seja acionada durante a contagem, será carregado automaticamente o novo valor do preset (PV) para a saída CV e o bloco funcional continua a contagem a partir do novo valor de CV.

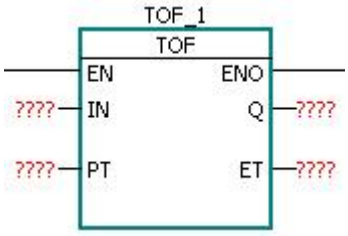
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Entrada **opcional**. Existente somente no modo com EN/ENO.
	CU	BOOL	Sinal de contagem (pulso).
	CD		
	R		Reset de contagem (CV = 0).
	LD		Carrega Preset (PV) em CV (efetivo).
	PV	INT, UINT, DINT e UDINT	Preset do contador.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Saída **opcional**. Existente somente no modo com EN/ENO.
	QU		Saída do contador crescente.
	QD		Saída do contador decrescente.
	CV	INT, UINT, DINT e UDINT	Efetivo do contador.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

Exemplo gráfico de funcionamento:



TOF - Temporizador: OFF Delay

O Bloco Funcional TOF pode ser utilizado de dois modos distintos: Com e sem EN/ENO. Cada modo possui uma característica diferente de trabalho e podem ser utilizados individualmente para cada TOF inserido no programa.



Modo: Com EN/ENO

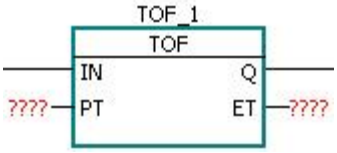
Com as entradas EN e IN habilitadas, na transição de TRUE para FALSE da entrada IN, inicia-se a contagem de tempo na saída ET (**Base de tempo 10ms**) até o valor de preset (PT). Durante essa contagem, a saída Q permanece ligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será desligada e permanecerá assim até que IN seja ligado. Quando a entrada IN é ligada, a saída Q é ligada novamente e o efetivo ET zerado (ET=0).

Se EN for desligado antes do fim da temporização, a temporização é PAUSADA, voltando ao normal quando EN for novamente habilitado.

O RESET de temporização acontece quando a entrada IN for ligada, somente se EN estiver habilitado. Do contrário o bloco não tem nenhum status alterado.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q desligada.



Modo: Sem EN/ENO

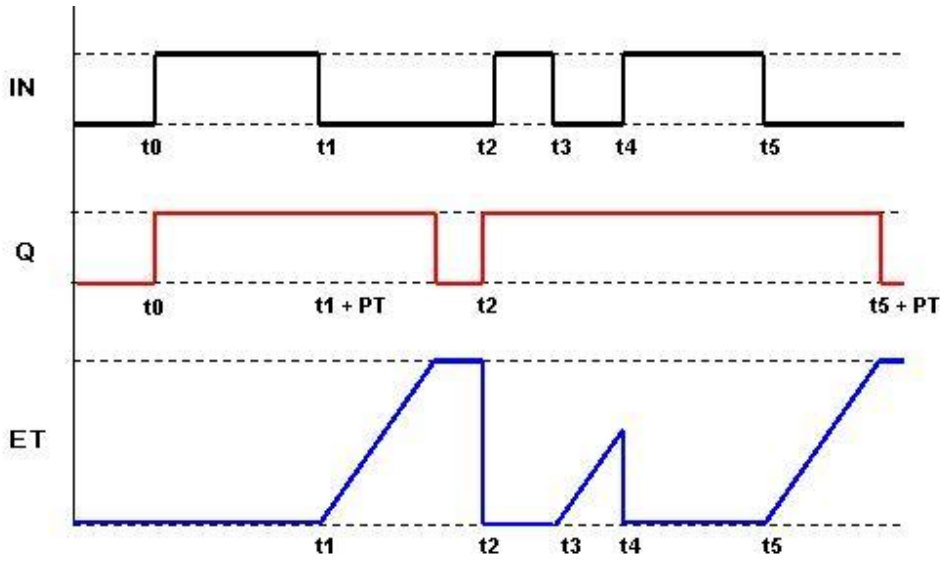
Com a entrada IN desabilitada, inicia-se a contagem de tempo na saída ET até o valor de preset (PT). Durante essa contagem, a saída Q permanece ligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será desligada e permanecerá assim até que IN seja ligado. Quando a entrada IN é ligada, a saída Q é ligada e o efetivo ET zerado (ET= 0).

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q desligada.

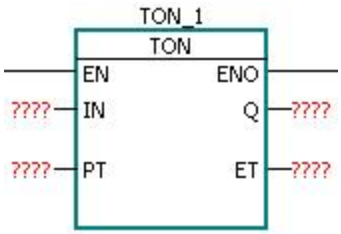
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Pausa temporização ao desabilitar antes do fim da contagem de tempo. Entrada **opcional**. Existente somente no modo com EN/ENO.
	IN	BOOL	Iniciar temporização.
	PT	TIME e CONSTANTE	Preset do temporizador.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Saída **opcional**. Existente somente no modo com EN/ENO.
	Q	BOOL	Desabilitado no fim da temporização
	ET	TIME	Efetivo do temporizador
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

Exemplo gráfico de funcionamento:



TON - Temporizador: ON Delay

O Bloco Funcional TON pode ser utilizado de dois modos distintos: Com e sem EN/ENO. Cada modo possui uma característica diferente de trabalho e podem ser utilizados individualmente para cada TON inserido no programa.



Modo: Com EN/ENO

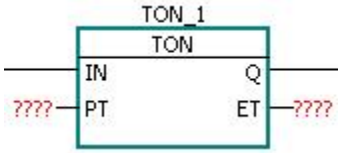
Com as entradas EN e IN habilitadas, inicia-se a contagem de tempo na saída ET (**Base de tempo 10 ms**) até o valor de preset (PT). Durante essa contagem, a saída Q permanece desligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será ligada e permanecerá assim até que IN seja desligado. Quando desligado, a saída Q é desligada e o efetivo ET zerado (ET= 0).

Se EN for desligado antes do fim da temporização, a temporização entra em PAUSE, voltando ao normal quando EN for novamente habilitado.

O RESET de temporização acontece quando a entrada IN for desligada, somente se EN estiver habilitado. Do contrário o bloco não tem nenhum status alterado.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q ligada.



Modo: Sem EN/ENO

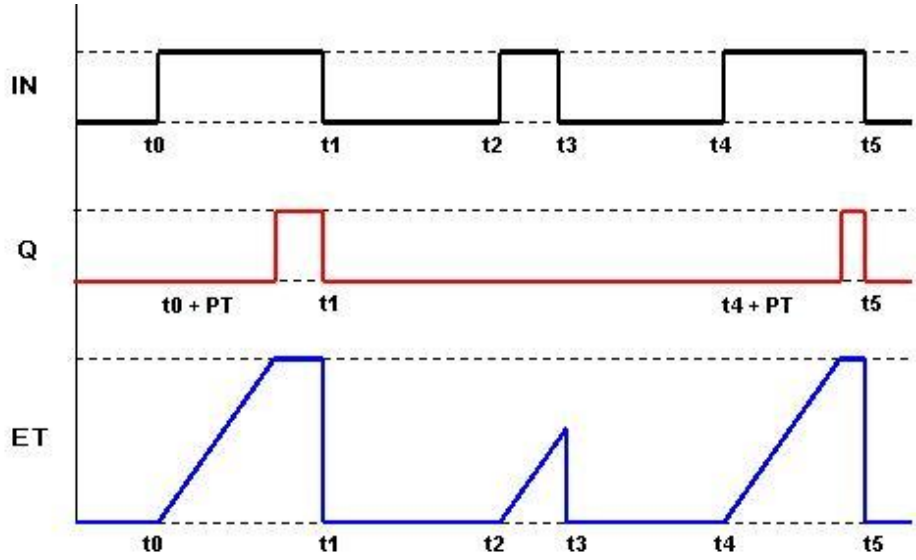
Com a entrada IN habilitada, inicia-se a contagem de tempo na saída ET (**Base de tempo 10 ms**) até o valor de preset (PT). Durante essa contagem, a saída Q permanece desligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será ligada e permanecerá assim até que IN seja desligado. Quando desligado, a saída Q é desligada e o efetivo ET zerado (ET= 0).

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q ligada.

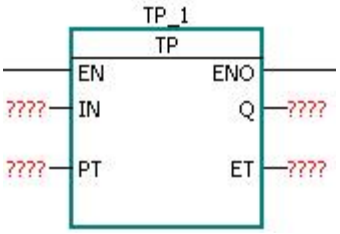
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Pausa temporização ao desabilitar antes do fim da contagem de tempo. Entrada **opcional**. Existente somente no modo com EN/ENO.
	IN	BOOL	Iniciar temporização
	PT	TIME e CONSTANTE	Preset do temporizador
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Saída **opcional**. Existente somente no modo com EN/ENO.
	Q	BOOL	Habilitado no fim da temporização
	ET	TIME	Efetivo do temporizador
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

Exemplo gráfico de funcionamento:



TP - Temporizador: Pulse mode

O Bloco Funcional TP pode ser utilizado de dois modos distintos: Com e sem EN/ENO. Cada modo possui uma característica diferente de trabalho e podem ser utilizados individualmente para cada TP inserido no programa.



Modo: Com EN/ENO

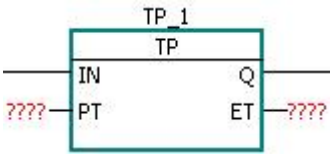
Com as entradas EN e IN habilitadas, inicia-se a contagem de tempo na saída ET (**Base de tempo 10ms**) até o valor de preset (PT). Durante essa contagem, a saída Q permanece ligada.

Permanecendo com a entrada IN habilitada, quando o efetivo (ET) for igual ao preset (ET=PT), a saída Q será desligada e o efetivo permanecerá com o valor final. O valor de ET será zerado (ET=0) somente quando a entrada IN for desligada. Ligando novamente a entrada IN, a saída Q é ligada e uma nova temporização iniciada.

Se EN for desligado antes do fim da temporização, a temporização entra em PAUSE, voltando ao normal quando EN for novamente habilitado.

Utilizando um pulso na entrada IN (ON e em seguida OFF), a saída Q será ligada e permanecerá assim até que se atinja a contagem do efetivo em ET. Quando ET=PT, a saída Q será desligada e o efetivo será zerado automaticamente, pois a entrada IN permaneceu desligada ao final da temporização.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q desligada.



Modo: Sem EN/ENO

Com a entrada IN habilitada, inicia-se a contagem de tempo na saída ET até o valor de preset (PT). Durante essa contagem, a saída Q permanece ligada.

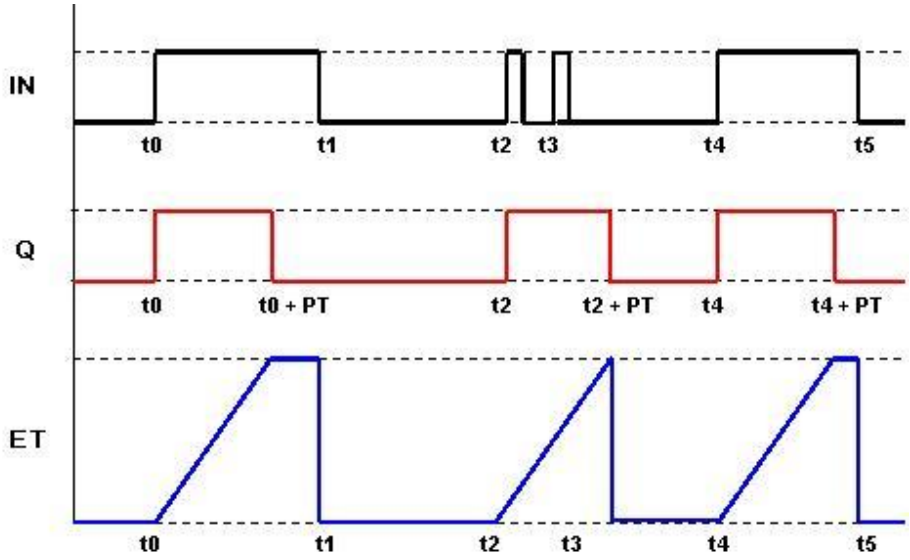
Permanecendo com a entrada IN habilitada, quando o efetivo (ET) for igual ao preset (ET=PT), a saída Q será desligada e o efetivo permanecerá com o valor final. O valor de ET será zerado (ET=0) somente quando a entrada IN for desligada. Ligando novamente a entrada IN, a saída Q é ligada e uma nova temporização iniciada.

Utilizando um pulso na entrada IN (ON e em seguida OFF), a saída Q será ligada e permanecerá assim até que se atinja a contagem do efetivo em ET. Quando ET=PT, a saída Q será desligada e o efetivo será zerado automaticamente, pois a entrada IN permaneceu desligada ao final da temporização.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Pausa temporização ao desabilitar antes do fim da contagem de tempo. Entrada **opcional**. Existente somente no modo com EN/ENO.
	IN	BOOL	Iniciar temporização.
	PT	TIME e CONSTANTE	Preset do temporizador.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Saída **opcional**. Existente somente no modo com EN/ENO.
	Q	BOOL	Habilitado no fim da temporização.
	ET	TIME	Efetivo do temporizador.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Exemplo gráfico de funcionamento:



Trigonométricas

ACOS - Função ARCO COSSENO

Quando a entrada EN for habilitada, será realizado o cálculo do arco cosseno de IN. O resultado é colocado na **saída da função**.

$$OUT = COS^{-1}(IN)$$

O resultado esta sempre entre os valores: -1 ≤ OUT ≤ 1.

Se não houver erro na execução da instrução, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL e CONSTANTE.	Variável de entrada (valor em radianos).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

ASIN - Função ARCO SENO

Quando a entrada EN for habilitada, será realizado o cálculo do arco seno de IN. O resultado é colocado na **saída da função**.

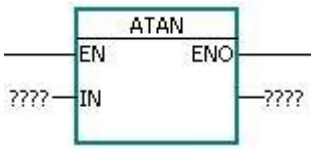
$$OUT = Sin^{-1}(IN)$$

O resultado estará sempre entre os valores: -1 ≤ OUT ≤ 1.

Se não houver erro na execução da instrução, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL e CONSTANTE.	Variável de entrada (valor em radianos).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

ATAN - Função ARCO TANGENTE



Quando a entrada EN for habilitada, será realizado o cálculo do arco tangente de IN. O resultado é colocado na **saída da função**.

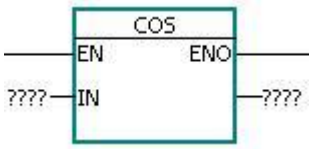
$$OUT = TAN^{-1}(IN)$$

O resultado estará sempre entre os valores: -1 ≤ OUT ≤ 1.

Se não houver erro na execução da instrução, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL e CONSTANTE.	Variável de entrada (valor em radianos).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

COS - Função COSSENO



Quando a entrada EN for habilitada, será realizado o cálculo do cosseno de IN. O resultado é colocado na **saída da função**.

$$OUT = COS(IN)$$

O resultado estará sempre entre os valores: -1 ≤ OUT ≤ 1.

Se não houver erro na execução da instrução, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL e CONSTANTE.	Variável de entrada (valor em radianos).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

SIN - Função SENO

Quando a entrada EN for habilitada, será realizado o cálculo do seno de IN. O resultado é colocado na **saída da função**.

$$OUT = Sin\,(IN)$$

O resultado estará sempre entre os valores: -1 ≤ OUT ≤ 1.

Se não houver erro na execução da instrução, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL e CONSTANTE.	Variável de entrada (valor em radianos).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

TAN – Função TANGENTE

Quando a entrada EN for habilitada, será realizado o cálculo da tangente de IN1. O resultado é colocado em OUT.

$$OUT = TAN\,(IN)$$

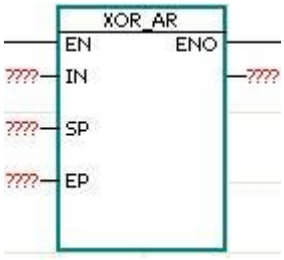
Se não houver erro na execução da instrução, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL e CONSTANTE	Variável de entrada (valor em radianos).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado.
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.
	Z	**Zero**	Será ligado se o resultado for ZERO.
	N	**Sinal**	Será ligado se o resultado for NEGATIVO.

CAPÍTULO 4
INTRUÇÕES ATOS

Array

XOR_AR



Com a entrada **EN** habilitada a instrução executa um XOR no array de bytes indicado na entrada **IN**.

A instrução possibilita realizar XOR em uma determinada região do array, ou seja, entre a posição inicial indicada na entrada **SP** e posição final na entrada **EP**.

O resultado é obtido na **saída da função**, onde deve ser um dado do tipo **BYTE** gerado no item ***Variáveis Globais***, ***grupo usuário*** e região de **DATA BLOCK**.

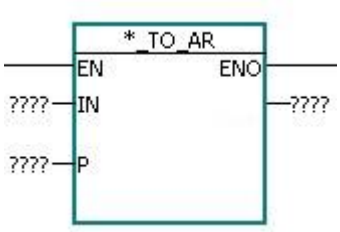
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	Array de bytes	Variável de entrada.
	SP	UINT	Posição inicial do array.
	EP		Posição final do array.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	BYTE	Resultado do XOR.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Configurando tipo de dado BYTE:

Exemplo de XOR_AR:

POSIÇÃO	VALOR	POSIÇÃO	VALOR
1	FF	7	FF
2	01	8	01
3	FF	9	FF
4	01	10	00
Resultado	00	Resultado	01

*_TO_AR



Quando a entrada de EN for habilitada, será armazenado o dado de IN na posição indicada por P no ARRAY indicada pela **saída da função**. A saída ENO será uma cópia da entrada EN.

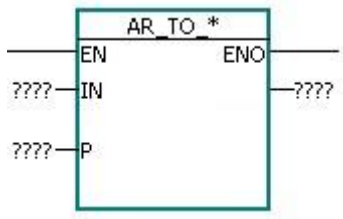
Quando a posição indicada por P for inválida, a saída ENO é desligada.

Essa instrução realiza o armazenamento de variáveis numa estrutura do Tipo Array.

O tipo da variável a ser armazenado deve ser do mesmo tipo do Array declarado.
O tamanho máximo do Array é de 65535 posições (UINT).

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN	INT, DINT, UINT, UDINT, WORD e DWORD, REAL, TIME, DATE, TOD, STRING e CONSTANTE	Variável cujo conteúdo é armazenado no ARRAY. O tipo de dado em IN deve ser igual ao tipo de dado do ARRAY declarado.
	P	UINT e CONSTANTE	Posição do ARRAY a inserir dado em IN
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Função	ARRAY	ARRAY de destino da variável IN
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

AR_TO_*



Quando a entrada de EN for habilitada, será lido o dado da posição indicada por P do Array de IN e armazenado na variável indicada pela **saída da função**. A saída ENO será uma cópia da entrada EN.
Quando a posição indicada por P for inválida, a saída ENO é desligada.

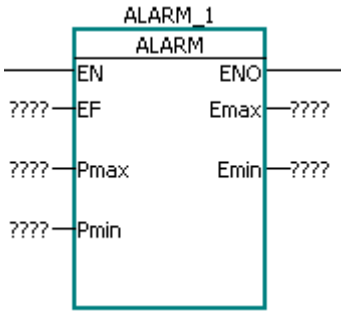
Essa instrução realiza a leitura de variáveis de um Array.
O tipo da variável a ser lido deve ser do mesmo tipo do Array declarado.

O tamanho máximo do Array é de 65535 posições (UINT).

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	ARRAY	ARRAY de origem dos dados.
	P	UINT e CONSTANTE	Posição do ARRAY a ser lida.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	OUT	INT, DINT, UINT, UDINT, WORD e DWORD, REAL, TIME, DATE, TOD, STRING e CONSTANTE	Variável para armazenar o valor lido do ARRAY. Deve ser utilizado um ARRAY de mesmo tipo de dado da variável IN.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Comparação

ALARM - Geração de alarmes



Esta função realiza a comparação de um valor de preset de máximo (Pmax) e mínimo (Pmin) com um efetivo declarado na entrada EF.

Caso a entrada Pmax ou a saída Emax permaneçam sem variável associada, será desprezada a comparação de máximo. Analogamente, se não existir a entrada Pmin ou a saída Emin será desprezada a comparação de mínimo.

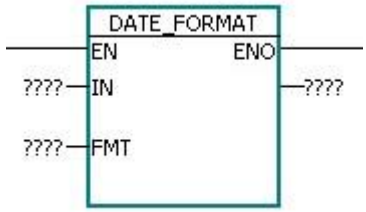
Quando a entrada EN for habilitada, será efetuada a comparação ligando ou desligando as saídas Emax e Emin de acordo com o critério:

- EF > ou = Pmax ➔ Emax = TRUE;
- EF < ou = Pmin ➔ Emin = TRUE.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	EF	INT, DINT, UINT, UDINT, WORD, DWORD, REAL e TIME	Efetivo de comparação. O tipo de dado desta entrada deve ser o mesmo de Pmax e Pmin.
	Pmax, Pmin	INT, DINT, UINT, UDINT, WORD, DWORD, REAL, TIME e ARRAY.	Variáveis de comparação de máximo e mínimo. O tipo de dado destas entradas deve ser o mesmo de EF. Caso seja utilizado um ARRAY, deve ser de mesmo tipo de EF e ter a mesma quantidade de posições de Emax e Emin.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Emax, Emin	BOOL e ARRAY	Variável booleana informando o resultado da comparação. Caso seja utilizado um ARRAY, deve ser do tipo BOOL e ter a mesma quantidade de posições do ARRAY utilizado nas entradas Pmax e Pmin.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Conversão

DATE_FORMAT



Essa instrução formata uma string de acordo com o formato dado na entrada FMT e o horário dado em IN. A string é gravada na **saída da função**.

As seguintes nomenclaturas são conhecidas para formatar a string:

%a - Dia da semana abreviado;

%A - Nome da semana completo;

%b - Nome do mês abreviado;

%B - Nome do mês completo;

%c - Representação da data e hora no formato: DDD MMM dd hh:mm:ss yyyy;

%C - Número do século (o ano dividido por 100 e truncado para um inteiro, de 00 até 99);

%d - Dia do mês como um número decimal (de 01 até 31);

%D - Mesmo que %m/%d/%y ;

%e - Dia do mês como um número decimal, um simples dígito é precedido por espaço (de ' 1' até '31');

%f - Representação da data e hora no formato yyyy-mm-dd ;

%H - Hora como um número decimal usando um relógio de 24-horas (de 00 até 23);

%I - Hora como um número decimal usando um relógio de 12-horas (de 01 até 12);

%j - Dia do ano como número decimal (de 001 até 366);

%k - Hora como um número decimal usando um relógio de 24-horas, um simples dígito é precedido por espaço (de ' 1' até '23');

%l - Hora como um número decimal usando um relógio de 12-horas, m simples dígito é precedido por espaço (de ' 1' até '12');

%m - Mês como número decimal (de 01 até 12);

%M - Minuto como número decimal (de 00 até 59);

%n - Caracter nova linha;

%p - Um dos dois 'AM' ou 'PM' de acordo com o valor da hora dada, ou as strings correspondentes para a localidade;

%P - Um dos dois 'am' ou 'pm' de acordo com o valor da hora dada, ou as strings correspondentes para a localidade;

%r - Hora em notação a.m. e p.m. (hh:mm:ss);

%R - Hora em notação de 24 horas (hh:mm);

%S - Segundo como um número decimal (00-59);

%t - Caracter "tab";

%T - Hora corrente, igual a %H:%M:%S ;

%w - Dia da semana como número decimal, Domingo sendo 0 (0-6);

%W - Número da semana do ano corrente como número decimal, começando com o Domingo como o primeiro dia da primeira semana (00-53);

%x - Representação preferida para a data mm/dd/yy (mesmo que %D) da localidade corrente, sem a hora;

%X - Representação preferida para a hora (hh:mm:ss) da localidade corrente, sem a data;

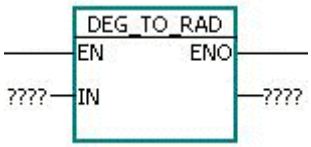
%y - Ano como número decimal, sem o século (de 00 até 99);

%Y - Ano como número decimal, incluindo o século (de 1970 até 2050);

%% - Caractere '%' literal;

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	DATE, TIME e DATE_AND_TIME	Tipo de horário do sistema a ser formatado.
	FMT	STRING	Formato da string para aplicar em IN.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	STRING	String IN formatada com os parâmetros definidos em FMT.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

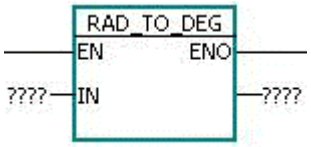
DEG_TO_RAD - Graus >> Radianos



Quando a entrada EN for habilitada, é feita a conversão do conteúdo de IN e colocado o valor na **saída da função**. Se não houver erro de conversão, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL	Valor em Graus.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado da conversão (em Radianos).
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.

RAD_TO_DEG - Radianos >> Graus

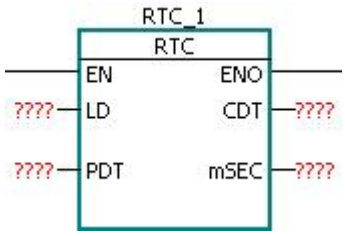


Quando a entrada EN for habilitada, é feita a conversão do conteúdo de IN1 e colocado o valor **saída da função**. Se não houver erro de conversão, a saída ENO será ligada, caso contrário será desligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	REAL	Valor em radianos.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	REAL	Resultado da conversão (em Graus).
	FLAG	**NOME**	**DESCRIÇÃO**
	OV	**Overflow**	Será ligado se houver estouro da variável de saída ou se a variável de entrada não for válida (NAN – Não é um número, por exemplo), caso contrário permanecerá sempre desligado.

Data / hora

RTC_ATOS – Real Time Clock



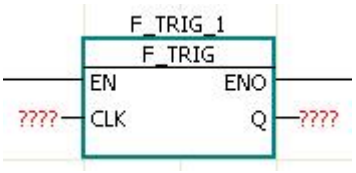
Esta instrução carrega e acerta data e hora do sistema (RTC).

Ela carrega a hora com exatidão de milisegundos.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	LD	BOOL	Quando habilitado, carrega PDT em CDT. Se desabilitado, mostra data e hora do sistema em CDT e mSEC.
	PDT	TOD, DATE e DATE_AND_TIME	Valor de entrada para ajuste do RTC.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	CDT	TOD, DATE e DATE_AND_TIME	Carrega data/hora atual do sistema.
	mSEC	UINT	Carrega milisegundos do sistema.
	FLAG	NOME	DESCRIÇÃO
	-------	-------	Nenhum *flag* é afetado.

Detecção de borda

F_TRIG – Detecta borda de descida

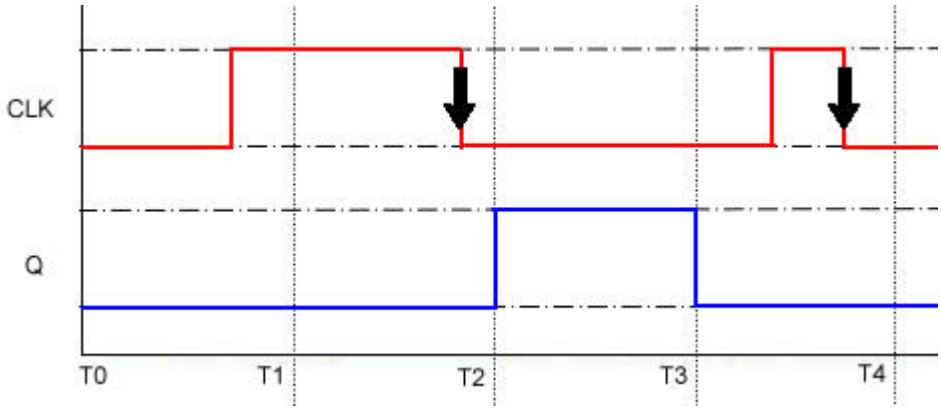


A instrução será executada sempre que a entrada EN estiver habilitada.

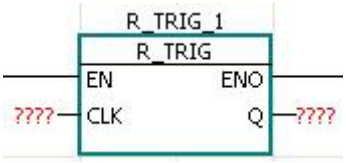
Durante a execução é verificado a alteração do estado de CLK de 1 para 0 (Borda de descida), caso ocorra esta alteração a saída Q é ligada durante uma varredura.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	CLK	BOOL	Variável de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Q	BOOL	Saída.
	FLAG	NOME	DESCRIÇÃO
	-------	-------	Nenhum *flag* é afetado.

Exemplo gráfico de funcionamento:



R_TRIG – Detecta borda de subida

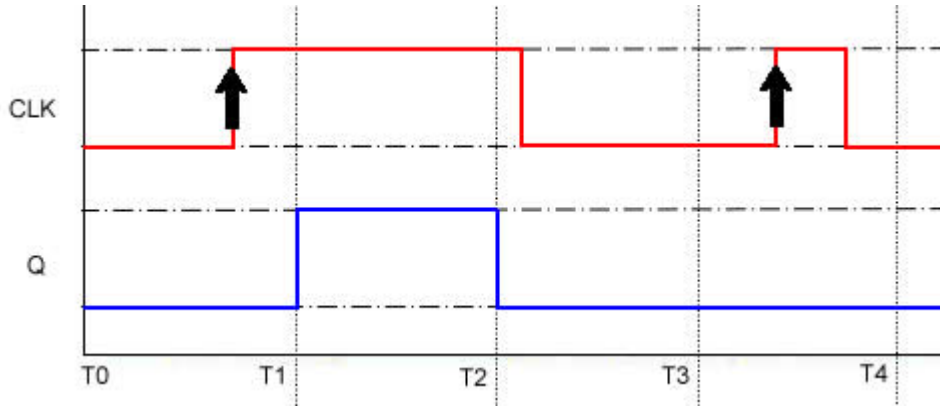


A instrução será executada sempre que a entrada EN estiver habilitada.

Durante a execução é verificado a alteração do estado de CLK de 0 para 1 (Borda de subida), caso ocorra esta alteração a saída Q é ligada durante uma varredura.

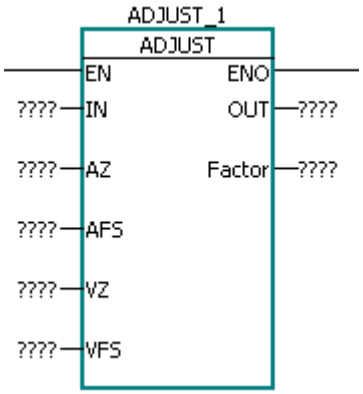
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	CLK	BOOL	Variável de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Q	BOOL	Saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	-------	-------	Nenhum *flag* é afetado.

Exemplo gráfico de funcionamento:



Especiais

ADJUST - Ajuste de escala



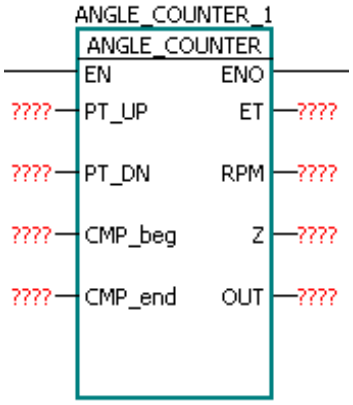
Quando a entrada EN for habilitada possibilita o ajuste linear da saída OUT em relação a uma variável definida pelo parâmetro IN. Quando EN estiver habilitada e a entrada AZ for habilitada, será feito um cálculo do fator de ajuste para o valor ZERO, caso a entrada AFS seja habilitada será feito o calculo do fator de ajuste para o FUNDO DE ESCALA. Esta instrução necessita da entrada de dois parâmetros VZ e VFS. Se houver erro no momento do calculo do fator de ajuste será ligado o flag de overflow.

O fator de ajuste é dado pela seguinte equação:
Fator = VFS – VZ
(Fundo de Escala Real - Zero Real)

O valor ajustado é dado pela seguinte equação:
Efetivo Ajustado=[(Efetivo Real - Zero Real) * FATOR] + Zero Relativo

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN	INT, UINT	Valor Efetivo Real
	AZ	BOOL	Habilita Cálculo do Fator de Ajuste do Valor Zero
	AFS	BOOL	Habilita Cálculo do Fator de Ajuste do Fundo de Escala
	VZ	INT, UINT	Valor do Zero Real
	VFS	INT, UINT	Valor do Fundo de Escala Real
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Factor	REAL	Fator de Ajuste
	OUT	INT, UINT	Valor Ajustado
	FLAG	**NOME**	**DESCRIÇÃO**
	Overflow	**OV**	É ligado se houver estouro no calculo do fator de ajuste
	Zero	**Z**	É ligado se o valor da saída for ZERO ou se no calculo do fator de ajuste houver divisão por ZERO.

ANGLE COUNTER



Contador modo Ângulo.

Esta instrução é vinculada ao hardware do controlador, portanto, é possível inserir somente uma instância deste Function Block no projeto.

Este Function Block não trabalha em conjunto com as instruções CPU COUNTER, HIGH SPEED COUNTER e ANGLE SIMULATE por utilizarem do mesmo recurso de hardware.

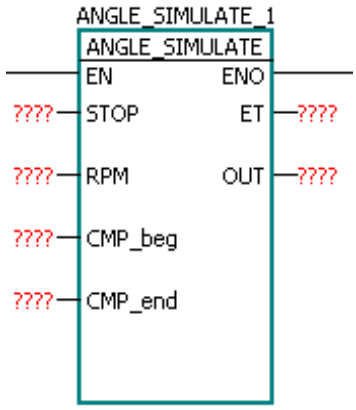
Observação: Utilizando este function block não será permitido adicionar ao programa as seguintes instruções:

- Angle_simulate;
- CPU_counter;
- HS_counter;

Atenção: As variáveis do tipo ARRAY devem ter no máximo 16 posições (exemplo: ARRAY [0..15] OF UINT ou ARRAY [1..16] OF UINT).

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	PT_UP	UINT e CONSTANTE	Preset de ZERO p/ contagem crescente
	PT_DN	UINT e CONSTANTE	Preset de ZERO p/ contagem decrescente
	CMP_beg	ARRAY (UINT)	Valores iniciais de comparação dos ângulos. Este ARRAY deve ser do mesmo tamanho dos ARRAYS definidos em CMP_end e OUT.
	CMP_end	ARRAY (UINT)	Valores finais de comparação dos ângulos. Este ARRAY deve ser do mesmo tamanho dos ARRAYS definidos em CMP_beg e OUT.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	ET	UINT	Efetivo de contagem
	RPM	UINT	Valor de RPM aferido na entrada de contagem
	Z	BOOL	Flag de Zero. Fica ligado por um scan.
	OUT	ARRAY (BOOL)	Resultado da comparação dos valores de CMP_beg e CMP_end com ET. Este ARRAY deve ser do mesmo tamanho dos ARRAYS definidos em CMP_end e CMP_beg.
	FLAG	NOME	DESCRIÇÃO
	-----	-----	Nenhum *flag* é afetado

ANGLE SIMULATE



Simulador de ângulo.

Esta instrução é vinculada ao hardware do controlador, portanto, é possível inserir somente uma instância deste Function Block no projeto.

Este Function Block não trabalha em conjunto com as instruções CPU_COUNTER, HIGH SPEED COUNTER e ANGLE_COUNTER por utilizarem do mesmo recurso de hardware.

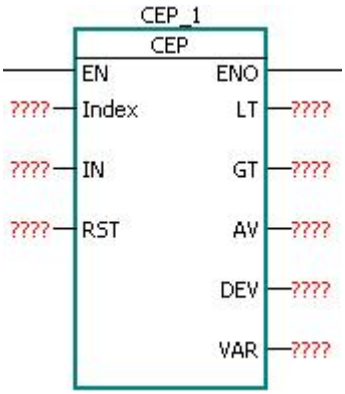
Observação: Utilizando este function block não será permitido adicionar ao programa as seguintes instruções:

- Angle_counter;
- CPU_counter;
- HS_counter;
- Stepmotor_fase;
- Stepmotor_pulse.

Atenção: As variáveis do tipo ARRAY devem ter no máximo 16 posições (exemplo: ARRAY [0..15] OF UINT ou ARRAY [1..16] OF UINT).

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	STOP	BOOL	Bloqueio de contagem do simulador
	RPM	UINT	Valor em RPM que o simulador deve contar Valor máximo de 500 RPM.
	CMP_beg	ARRAY (UINT)	Valores dos ângulos iniciais de comparação. Este ARRAY deve ser do mesmo tamanho dos ARRAYS definidos em CMP_end e OUT.
	CMP_end	ARRAY (UINT)	Valores dos ângulos finais de comparação. Este ARRAY deve ser do mesmo tamanho dos ARRAYS definidos em CMP_beg e OUT.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	ET	UINT	Efetivo de contagem
	OUT	ARRAY (BOOL)	Saídas de comparação dos ângulos. Este ARRAY deve ser do mesmo tamanho dos ARRAYS definidos em CMP_end e CMP_beg.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

CEP – Controle Estatístico do Processo



A instrução será executada sempre que a entrada EN estiver habilitada.

Esta instrução quando habilitada, executa uma série de operações estatísticas com método de trabalho do tipo **FIFO** (***First In First Out***).

EN

A entrada deve ser habilitada com pulso de transição positiva (False para True), pois a cada pulso executado um dado da entrada **IN** será armazenado numa posição da array e será direcionado para próxima posição da array.

OBS.: Caso a entrada EN fique em True constante, todas as posições da array serão presetadas com o valor da entrada IN.

Index

Esta entrada deve ser um array do tipo INT ou UINT configurada com até **128 posições**. Para configurar um array segue:

1. Selecionar ícone **Global Variables** na **Área do Gerenciador do Projeto**;
2. Selecionar com duplo clique do mouse o ícone **User** (Variáveis de usuário);
3. Selecionar tipo de dado **DATA BLOCK** no campo **Região do Data Type**;
4. No campo **Tipo de dado** inserir ARRAY[**n..m**] OF INT ou ARRAY[**n..m**] OF UINT.

Onde **n** é o número de início e **m** o número de final da array.

IN

O tipo de dado da entrada **IN** deve ser o mesmo que o tipo de dado da array.

Para inserir um dado, é necessário habilitar a entrada **EN** com um pulso de transição positiva. Automaticamente é retornado todos os cálculos estatísticos sobre o novo dado.

RST - Reset

Esta entrada zera todos os dados das posições da array, saídas e volta para a primeira posição da array.

Para executar o reset:

1. **RST** = TRUE
2. **EN** = TRUE

LT - Menor valor

Dentre todos os valores das posições na array, a saída LT mostra o **menor** deles.

GT - Maior valor

Dentre todos os valores das posições na array, a saída GT mostra o **maior** deles.

AV - Média

A saída **AV** mostra o resultado do cálculo de média das amostras.

A média é obtida dividindo-se a soma dos valores de cada posição pelo número total de posições.

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

Onde **Xi** é o valor em cada posição e **n** o número de posições.

OBS.: O valor da média pode ser arredondado quando utilizamos um campo de visualização na IHM. Este arredondamento depende da precisão configurada no campo Edit/View.

DEV - Desvio padrão amostral

O **desvio padrão** é a medida mais comum da dispersão estatística. O desvio padrão define-se como a raiz quadrada da **variância**. É definido desta forma de maneira a dar uma medida da dispersão que seja um número não negativo.

A letra **S** representa o desvio padrão da amostra.

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2}$$

OBS.: O valor do desvio padrão amostral pode ser arredondado quando utilizamos um campo de visualização na IHM. Este arredondamento depende da precisão configurada no campo Edit/View.

VAR - Variância amostral

A variância de uma variável aleatória é uma medida da sua dispersão estatística (Desvio padrão amostral), indicando quão longe em geral os seus valores se encontram do valor esperado.

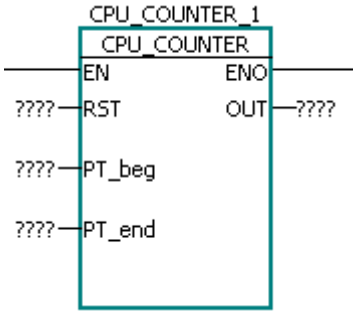
$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \overline{X}\right)^2$$

OBS.: O valor da variância amostral pode ser arredondado quando utilizamos um campo de visualização na IHM. Este arredondamento depende da precisão configurada no campo Edit/View.

CEP – (Continuação)

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Usar pulso de transição positiva (False para True) ativado durante uma varredura (MONOA).
	Index	ARRAY do tipo INT e do tipo UINT	Endereço da Array.
	IN	INT e UINT	Dado a ser inserido na posição da array.
	RST	BOOL	Zera todos os dados da array e das saídas. Volta para a primeira posição da array.
SAÍDA	ENO		Cópia do valor booleano de EN.
	LT	INT e UINT	Menor valor.
	GT		Maior valor.
	AV	REAL	Média amostral.
	DEV		Desvio Padrão Amostral.
	VAR		Variância Amostral.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

CPU COUNTER - Contador da CPU



Contador da CPU.

A contagem de pulsos é feita por meio da entrada física do CLP E0 (**%I0.0**) para contagem **Unidirecional** ou E0 (**%I0.0**) e E1 (**%I0.1**) para contagem **Bidirecional** (**Ver jumpers de configuração**).

A saída OUT fica acionada enquanto o valor do EFETIVO estiver entre PT_beg e PT_end.

Todos os Function Blocks CPU_COUNTER declarados no projeto utilizam o mesmo efetivo para contagem:

%SD10: EF_HS_COUNTER

É possível inserir até 8 instâncias deste Function Block no projeto.
Por utilizarem do mesmo recurso de hardware, os Function Blocks: HIGH SPEED COUNTER, ANGLE COUNTER e ANGLE SIMULATE não podem ser inseridos num projeto que já utilize a instrução CPU COUNTER.

Observação: Utilizando este function block não será permitido adicionar ao programa as seguintes instruções:

- Angle_counter;
- Angle_simulate;
- HS_counter;

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	RST	BOOL	Reset do efetivo de contagem (pulso)
	PT_beg	DINT	Preset inicial
	PT_end	DINT	Preset final
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	OUT	BOOL	Saída do hardware da CPU (%Q0.0 a %Q0.7). Mantém-se acionada enquanto valor do efetivo estiver entre PT_beg e PT_end.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

Jumpers de configuração

Jumpers da CPU MPC4004BF Contador Rápido	
ST2 - A	Unidirecional
ST3 - B	
ST2 - B	Bidirecional
ST3 - A	

A CPU MPC4004BF é composta por duas placas de circuito impresso. Os jumpers encontram-se na placa inferior (101.2806 "P" ou 101.2807 "N").

EVENT_LOG – Log de eventos

A instrução será executada sempre que a entrada **EN** estiver habilitada.

Seleção do modo de operação

A instrução gera um log de eventos armazenado em um array indicado pela entrada **ARRAY**. Possibilita dois métodos de operação das pilhas dentro do array: **FIFO** e **LIFO**.

Esta seleção entre FIFO ou LIFO é dada pela entrada **MODE**.

Pilha de dados no array modo FIFO

- **FIFO** (First In First Out) - Primeiro dado inserido no array será o primeiro dado a ser removido quando a entrada REM (remover para a saída OUT) for igual a True;

Exemplo para um array de 5 posições:

Ordem de entrada dos dados	Posição do Array	Ordem de saída dos dados
PRIMEIRO	→ [1] →	PRIMEIRO
SEGUNDO	→ [2] →	SEGUNDO
TERCEIRO	→ [3] →	TERCEIRO
QUARTO	→ [4] →	QUARTO
QUINTO	→ [5] →	QUINTO

Pilha de dados no array modo LIFO

- **LIFO** (Last In First Out) - Último dado inserido no array será o primeiro dado a ser removido quando a entrada REM (remover para a saída OUT) for igual a True.

Exemplo para um array de 5 posições:

Ordem de entrada dos dados	Posição do Array	Ordem de saída dos dados
PRIMEIRO	→ [1] →	QUINTO
SEGUNDO	→ [2] →	QUARTO
TERCEIRO	→ [3] →	TERCEIRO
QUARTO	→ [4] →	SEGUNDO
QUINTO	→ [5] →	PRIMEIRO

Inserindo dados no array

Com a entrada **EN** habilitada, a cada pulso na entrada **ADD** o conteúdo de IN será copiado para uma posição do array. **O array deve ser do mesmo tipo de dado da entrada IN.**

A saída CNT indicada o número de amostras inseridas no array, ou seja, o número de posições ocupadas. Caso um valor seja removido para a saída OUT este valor será decrementado. O valor máximo para a saída CNT exibir será o mesmo número de posições no array.

Conforme os pulsos em ADD a saída CNT é encrementada e quando seu valor corresponder ao número total de posições a saída ENO será desligada e não será possível inserir nenhum outro valor no array até que seja removido um valor ou mesmo resetado o conteúdo do array.

Copiando / movendo dados do array para a saída OUT

Para copiar um determinado valor do log para a saída sem retirar o mesmo da pilha, deve ser informado a posição do dado dentro do array por meio da entrada **Index** e em seguida habilitar a entrada **GET**. A saída **CNT** permanece inalterada.

Antes de mover um dado da pilha do array deve-se lembrar do modo de operação da instrução, ou seja, verificar o status de acionamento da entrada **MODE**.

Para remover um determinado valor do log para a saída OUT, basta habilitar a entrada **REM**. Com a entrada MODE em TRUE (FIFO) será removido o primeiro dado inserido no array, se estiver em FALSE será removido o último dado inserido no array. A saída **CNT** é alterada para um novo valor correspondente ao número de posições ainda ocupadas.

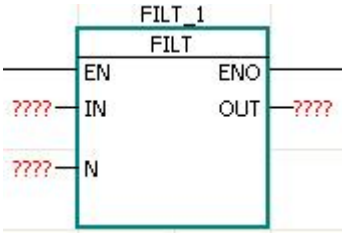
Observação: embora existam as opções de **GET** e **REM** na instrução, qualquer elemento do array pode ser visualizado e manipulado por outras instruções, inclusive supervisionado via A1 ou software supervisório, pois a pilha que representa o log de evento é um array declarado dentro da região de tipo de dado **DATA BLOCK** em ***Variáveis Globais*** de usuário.

Reset do array

Com a entrada **RST** em TRUE todo o conteúdo do array será resetado, inserindo FFFFh em todas as posições, zerando também o contador CNT.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	Todos os tipos de dados incluindo Struct.	Tipo de dado de entrada ou estrutura de dados criada pelo usuário.
	RST	BOOL	Reseta o conteúdo do array (insere 0xFFFF) e zera a saída CNT.
	REM		Remove dado do array, permitindo que um novo valor seja inserido.
	ADD		Insere o conteúdo de IN no array.
	MODE		Modo de operação da pilha de informações no array: • TRUE = FIFO; • FALSE= LIFO.
	ARRAY	ARRAY	Armazena o log de eventos. Array deve ser do mesmo tipo de dado da entrada **IN**.
	Index	UINT	Apontador do índice no array. Seleciona uma posição dentro do array. Funciona somente para indicar a posição do array no qual terá seu conteúdo copiado para a saída **OUT** quando GET igual a TRUE.
	GET	BOOL	Copia dado da posição indicado pela posição **Index** para a saída **OUT** sem mover o dado do array.
SAÍDA	ENO	BOOL	Desliga quando o valor da entrada Index for maior que o número de posições do array ou quando todas as posições do array estão ocupadas.
	OUT	Todos os tipos de dados incluindo Struct.	Recebe o conteúdo de uma posição do array. Deve ser do mesmo tipo de dado da entrada **IN**.
	CNT	UINT	Exibe o número de posições do array com dados inseridos. Incrementa quando é inserido um dado da entrada IN no array. Este valor é incrementado até o limite de posições do array, por exemplo, Array de 5 posições o valor máximo de CNT será igual a 5. Decrementa quando um dado é removido de uma posição do array.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum flag é afetado.

FILT - Filtro



Classifica um universo entre 3 a 31 amostras de uma variável de entrada, obtendo como resultado o valor da amostra central.

Quando a entrada **EN** for acionada (pulso), o valor contido na entrada **IN**, será capturado como nova amostra e substituirá a amostra mais antiga. Em seguida será executada a classificação das amostras (valores das amostras em ordem crescente) e o valor da amostra central, após a classificação, será colocado na saída **OUT** (variável de saída do filtro).

A primeira execução da instrução FILT (após inicialização do CLP) preenche a pilha de amostras (buffer) e o valor da variável de saída **OUT** com o primeiro valor lido da variável de entrada **IN**.

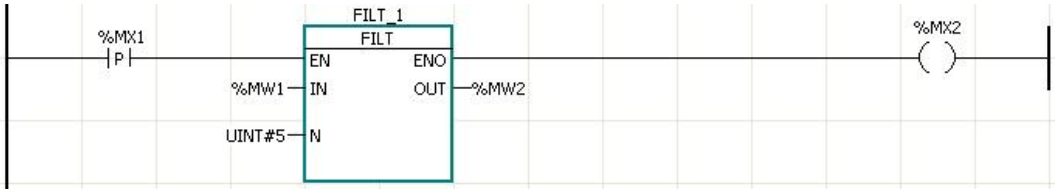
O BUFFER de armazenamento das amostras pode conter até 31 elementos na pilha, sempre descartando o valor mais antigo da pilha e colocando em ordem crescente os valores.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Utilizar acionamento por pulso.
	IN	INT, DINT, UINT, UDINT, WORD e DWORD, REAL, TIME, DATE, TOD e DATE_AND_TIME	Variável de entrada do filtro.
	N	Constante do tipo UINT	Número (**ímpar**) de amostras no buffer do filtro.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Será desligado caso a entrada **N** contenha um valor par, um valor menor que 3 ou maior que 31.
	OUT	INT, DINT, UINT, UDINT, WORD e DWORD, REAL, TIME, DATE, TOD e DATE_AND_TIME	Variável de saída do filtro. Valor da amostra central.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum flag é afetado.

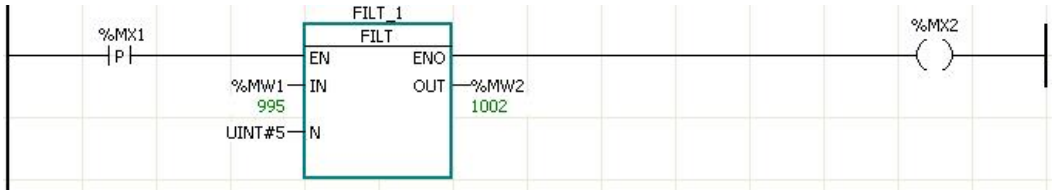
Exemplo de FILT

Quando a entrada **EN** for acionada através de um pulso, o valor contido em **IN** (exemplo %MW1) será capturado para ser classificado com mais 4 amostras previamente aquisitadas, conforme o número de amostras configurado em N.

Após a classificação o valor central das amostras será colocado no conteúdo da variável de saída **OUT** (exemplo %MW2). A região onde as amostras serão armazenadas e classificadas não é disponível ao usuário e a execução da instrução pode ser exemplificada da seguinte forma:

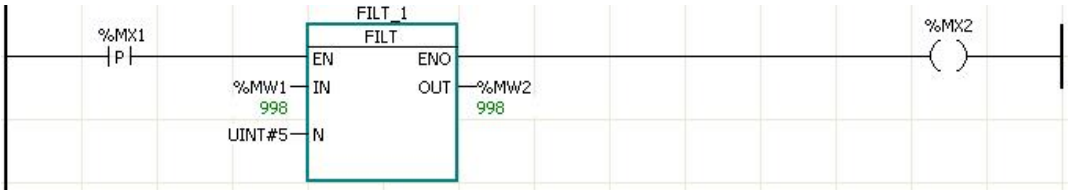


		Amostra	Valor		Valor		Saída OUT	
	Amostra mais antiga	N - 4	1034		995		Variável	Valor
		N - 3	998	Classificação	998			
Entrada IN		N - 2	1002		1002	Valor central	%MW2	1002
Variável	Valor	N - 1	1018		1018			
%MW1	995	Amostra mais recente	N	995	1034			



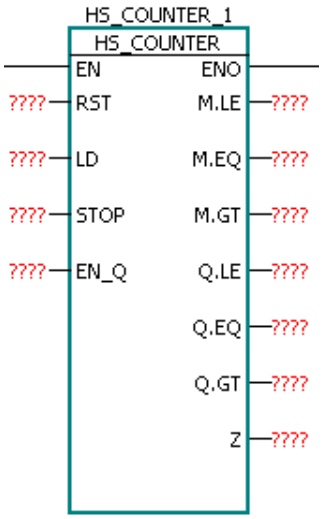
Após o acionamento da entrada **EN**:

		Amostra	Valor		Valor		Saída OUT	
	Amostra mais antiga	N - 4	998		995		Variável	Valor
		N - 3	1002	Classificação	998			
Entrada IN		N - 2	1018		998	Valor central	%MW2	998
Variável	Valor	N - 1	995		1002			
%MW1	998	Amostra mais recente	N	998	1018			



O conteúdo do BUFFER apontado pela entrada **N** não expressa a classificação das amostras, e para se conhecer a amostra mais antiga ou a recente, há necessidade de cálculo juntamente com o contador da pilha, pois uma amostra não é remanejada de sua posição e sim descartada (a mais antiga) quando uma nova amostra é adquirida e colocada em seu lugar (método de pilha circular). O valor do contador determina o deslocamento a ser somado ao endereço da primeira amostra (OFF-SET) afim de apontar para o endereço da próxima amostra a ser descartada e atualizada.

HIGH SPEED COUNTER



As saídas associadas ao hardware do controlador (Q.LT, Q.EQ e Q.GT) só serão atualizadas se a entrada **EN_Q** estiver acionada, caso contrário permanecem sempre desligadas.

Observação: Utilizando este function block não será permitido adicionar ao programa as seguintes instruções:

- Angle_counter;
- Angle_simulate;
- CPU_counter;

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	RST	BOOL	Reset do efetivo do contador rápido
	LD	BOOL	Carrega valor inicial do efetivo de contagem, definido na variável de sistema %SD12 (INI_HS_COUNTER)
	STOP	BOOL	Bloqueio de contagem
	EN_Q	BOOL	Habilita atualização das saídas Q.LT, Q.EQ e Q.GT no hardware do controlador.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	M.LE	BOOL	Status de comparação: efetivo **<** preset
	M.EQ	BOOL	Status de comparação: efetivo **=** preset
	M.GT	BOOL	Status de comparação: efetivo **>** preset
	Q.LE	BOOL	Saída de comparação: efetivo **<** preset. Sua declaração é opcional. (deve ser uma saída digital da CPU)
	Q.EQ	BOOL	Saída de comparação: efetivo **=** preset. Sua declaração é opcional. (deve ser uma saída digital da CPU)
	Q.GT	BOOL	Saída de comparação: efetivo **>** preset. Sua declaração é opcional. (deve ser uma saída digital da CPU)
	Z	BOOL	ON quando efetivo = zero
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

PID - Algoritmo PID ISA

A instrução PID incorpora recursos e características, necessárias para os seguimentos de processos contínuos:

- Modo Direto / Reverso
- Anti-reset ou Anti-windup
- Modo Manual / Automático
- Zona Morta
- Feedforward-BIAS: Offset da saída de controle
- Limites de Saída
- Tempo de Amostragem (Dt)

A equação PID adotada, segue o padrão ISA, conforme a equação mostrada abaixo:

$$S = K\cdot\left(e_{(t)} + Ki\cdot\int edt \;+ T_d\cdot de/dt\right) + BIAS$$

OU

$$S = K\cdot\left(e_{(t)} + \sum_{0}^{\infty} Ki\cdot e_{(t)}\cdot\Delta t + T_d\cdot\Delta e/\Delta t\right) + BIAS$$

sendo: Ki = 1/Ti

Descrição da nomenclatura utilizada:

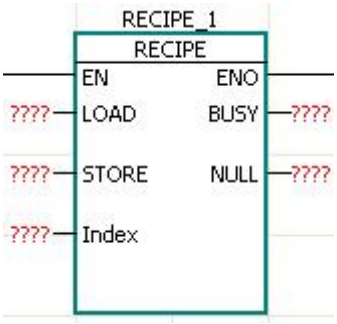
- K = ganho do sistema pois multiplica os termos P,I e D
- Ki = ganho integral
- Td= ganho derivativo (tempo do termo derivativo)
- dt = Tempo de amostragem
- de = (Erro atual – Erro anterior)
- Ti = 1 / Ki = Tempo do termo Integral (reset time)
- BIAS = Offset da saída de controle

Mais informações sobre o algoritmo desta instrução e métodos de sintonia "on-line" podem ser encontrados no manual "**Algoritmo PID padrão ISA**", disponível para download gratuitamente no site da Atos.

PID - Algoritmo PID ISA (continuação)

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	MAN	BOOL	Habilita controle PID em modo manual. TRUE = MODO MANUAL FALSE = MODO AUTOMÁTICO
	LD_I	BOOL	Carrega valor do termo integral definido em I_VALUE.
	DIR	BOOL	Define modo: TRUE = DIRETO FALSE = REVERSO
	PV	INT e UINT	Variável de entrada (ex: canal de temperatura). Deve ser do mesmo tipo de dado de ST e MV.
	ST	INT e UINT	Variável de Preset. Deve ser do mesmo tipo de dado de PV e MV.
	PERIOD	UINT	Período de amostragem.
	K	UINT	Ganho proporcional.
	TI	UINT	Ganho integral.
	TD	UINT	Ganho derivativo.
	MV_BIAS	INT	Offset de saída.
	DB_HIGH	INT	Banda morta alta.
	DB_LOW	INT	Banda morta baixa.
	PERIOD_PWM	UINT	Tempo do PWM da saída MV (período).
	L_PV_HIGH	INT	Máximo valor de escala para entrada.
	L_PV_LOW	INT	Mínimo valor de escala para entrada.
	L_MV_HIGH	INT	Máximo valor de escala para saída.
	L_MV_LOW	INT	Mínimo valor de escala para saída.
	I_VALUE	INT	Valor de carga do termo integral (TI).
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	MV	INT e UINT	Variável de saída. Deve ser do mesmo tipo de dado de PV e ST.
	PWM	BOOL	Variável de saída PWM (booleana).
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

RECIPE – Arquivo de receitas



A instrução será executada sempre que a entrada EN estiver habilitada.

É permitido um bloco funcional RECIPE por projeto.

Arquivo de Receitas, é utilizado em aplicações onde existe a necessidade de se carregar, em endereços pré-determinados, conjuntos de parâmetros previamente configurados e armazenados na memória do CLP.

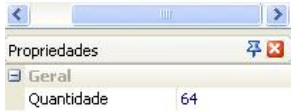
Essa função realiza o armazenamento ou a recuperação de até **16 frames** de dados. **O conjunto de frames programados formará uma Receita.**

Exemplo: Conjunto de até 16 frames configurados formado uma receita.

O usuário tem disponível 32 kbytes na memória para arquivo de receitas.

Selecionando o ícone **Receitas**

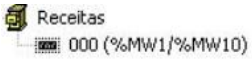
a guia de ***Propriedades*** apresentará um campo **Quantidade** onde é editado o número de receitas.



Configuração dos Frames

Para adicionar um novo Frame e configurá-lo, basta selecionar com o cursor do mouse o ícone **Recipes** localizado na guia ***Gerenciador do Projeto***, clicar com o botão direito e selecionar a opção **New Frame** (Máximo 16 Frames). Se desejar **excluir o Frame**, clique com o botão direito do mouse no ícone do Frame e selecione a opção **Apagar**.

New Frame



Em cada frame, identifique o primeiro e o último endereço das variáveis que serão utilizadas, respectivamente nos campos Inicial e Final.

Armazenar Receita (STORE)

Para armazenar uma receita segue a seqüência:

1. Indicar o número da receita na entrada Index (não pode ser zero);
2. Definir valores para as variáveis configuradas no(s) frame(s);
3. Setar STORE = TRUE.

Receita Ocupada

Caso a receita a ser armazena estiver ocupada, a saída BUSY será setada para TRUE.

Para uma receita ocupada há duas possibilidades de ação: Sobrepor a receita existente ou não armazenar a receita e configurar outro número na entrada Index.

Sobrepor receita

1. Manter STORE = TRUE (Saída BUSY=TRUE);
2. Setar BUSY = FALSE (por meio de lógica no programa);
3. Setar STORE = FALSE.

Não sobrepor receita

1. Setar STORE = FALSE;
2. Setar BUSY = FALSE.

Receita nula

Caso seja indicado um número na entrada Index não correspondente à quantidade de receitas configuradas, a saída NULL será setada para TRUE, tanto para armazenar (STORE) quanto para carregar (LOAD) uma receita.

Para setar a saída NULL = FALSE, desabilite a instrução (EN = FALSE) ou utilize lógica no programa.

Carregar Receita (LOAD)

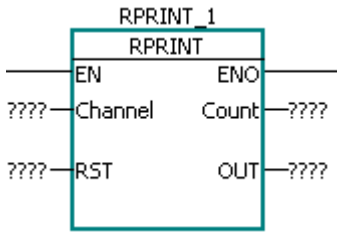
Para carregar uma receita segue a seqüência:

1. Indicar o número da receita na entrada Index (não pode ser zero);
2. Setar LOAD = TRUE (pode ser um pulso);
3. Os valores para as variáveis configuradas no(s) frame(s) serão carregados.

RECIPE(Continuação)

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	LOAD		Carrega receita.
	STORE		Armazena receita.
	Index	UINT, DINT, UDINT, WORD e DWORD	Endereço para armazenar receita.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	BUSY		Sinaliza receita ocupada.
	NULL		Sinaliza receita nula.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

RPRINT – Leitura de canal serial



A instrução será executada sempre que a entrada **EN** estiver **habilitada**.

Esta instrução quando habilitada recebe dados pelo canal serial e aloca em uma STRING ou Array de bytes.

Antes de executar a instrução, acionando a entrada EN, é **necessário acionar a variável de sistema** que dispõe o canal serial para recepção de dados. Para o **canal A (RS232)** habilitar a variável de sistema **READER_A** e para o **canal B (RS485)** habilitar a variável de sistema **READER_B**.

Enquanto a variável de sistema (READER_A ou READER_B) que habilita o canal serial estiver acionada, o canal de comunicação serial ficará indisponível para supervisão e envio de programa. Desta forma, a variável deve possuir uma lógica associada para ser habilitada somente durante a execução da instrução.

Recebendo uma quantidade de dados no canal serial maior que o tamanho da variável de saída OUT declarada, os dados que excedam o tamanho da variável serão armazenados em um buffer interno do CLP. A saída OUT irá armazenar dados até o limite declarado e a saída ENO será desligada. Para armazenar os dados na variável de saída OUT que ainda estão no buffer é necessário limpar o conteúdo de OUT habilitando a entrada RST ou movendo para outro endereço, pois OUT deve estar vazia para continuar recebendo dados.

Exemplo:

OUT= String[10] - pode armazenar até 10 caracteres.

Pelo canal serial foram enviados 36 caracteres:

1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ

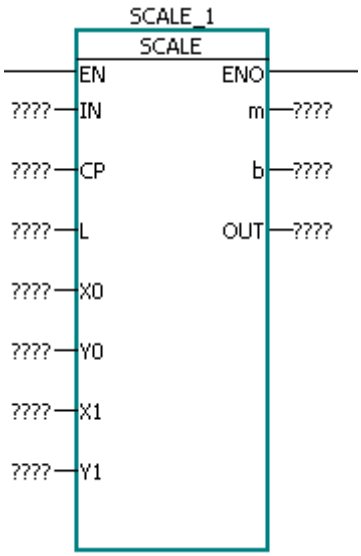
Para o recebimento destes dados segue:

1. OUT=1234567890 - Count=10 e ENO=FALSE;
2. RST=TRUE (pulso);
3. OUT=ABCDEFGHIJ - Count=10 e ENO=FALSE;
4. RST=TRUE (pulso);
5. OUT= KLMNOPQRST - Count=10 e ENO=FALSE;
6. RST=TRUE (pulso);
7. OUT= UVWXYZ - Count=6 e ENO=TRUE.

RPRINT(Continuação)

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	Channel	UINT	Seleciona canal de comunicação: 0 - RS232 1 - RS485
	RST	BOOL	Limpa conteúdo da saída OUT e zera a saída Count. Deve ser acionado por pulsos.
SAÍDA	ENO	BOOL	Será desligado quando: • Entrada Channel for difere de 0 ou 1; • Count for igual a quantidade de caracteres/bytes declarados na variável de saída OUT.
	Count	UINT	Quantidade de caracteres/bytes recebidos pelo canal serial.
	OUT	STRING Array de byte	Armazena o valor recebido via serial.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

SCALE - Ajuste de escala



Gera uma saída linear em relação à entrada.
Os parâmetros de entrada são:

x0 = valor inicial do ponto x
y0 = valor inicial do ponto y
x1 = valor final do ponto x
y1 = valor final do ponto y

A fórmula para conversão da entrada é: **OUT = m*IN + b**
onde: **OUT** = saída
m = fator de escalonamento **(Y1 - Y0) / (X1 - X0)**
IN = entrada
b = offset (**b = Y1 - m*X1**)

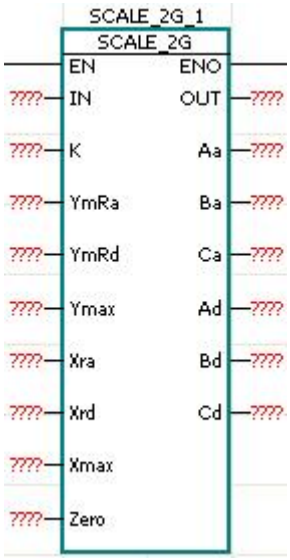
Se as entradas **EN** e **CP** estiverem ligadas, será feito o cálculo do valor de **m**, antes de calcular **OUT**, caso a entrada **CP** estiver desligada então será feito o cálculo de **OUT** com o último valor de **m** calculado.

A instrução define limites para o valor de **OUT**, quando a entrada **L** estiver habilitada. O limite será obtido através dos próprios valores de **Y0** e **Y1**, fornecidos pelo usuário, garantindo assim com que os valores de **OUT** estejam sempre dentro dos limites dados por **Y0** e **Y1**.

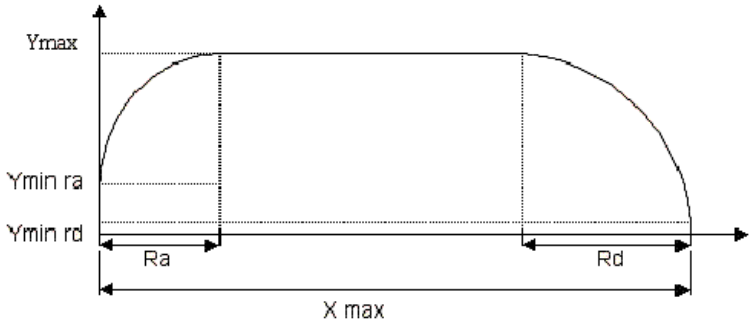
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN	INT, DINT, UINT, UDINT, REAL e TIME	Valor Efetivo da Entrada
	CP	BOOL	Habilita Cálculo dos Fatores "m" e "b"
	L	BOOL	Habilita Cálculo dos Limites da Saída
	X0	INT, DINT, UINT, UDINT, REAL, TIME e CONSTANTE	Valor Inicial da Abscissa X
	Y0		Valor Inicial da Ordenada Y
	X1		Valor Final da Abscissa X
	Y1		Valor Final da Ordenada Y
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	m	REAL	Fator de Escalonamento
	b	REAL	Fator de Offset
	OUT	INT, DINT, UINT, UDINT, REAL e TIME	Valor da Saída Calculada

FLAG	NOME	DESCRIÇÃO
Overflow	**OV**	Será ligado se houver estouro de variável.
Zero	**Z**	Será ligado se o resultado de **b** ou da saída **OUT** for zero.
Sinal	**S**	Será ligado se valor de **OUT** for negativo.

SCALE2G - Escala de 2º grau



Esta instrução é utilizada para gerar rampas de aceleração e desaceleração. Se a entrada EN estiver habilitada e a entrada K estiver desabilitada, a instrução gerará em OUT uma saída que segue uma função do segundo grau em relação à entrada IN. Se a entrada K for habilitada enquanto EN estiver habilitada, será feito o cálculo dos coeficientes de ajuste para as equações de aceleração e desaceleração.



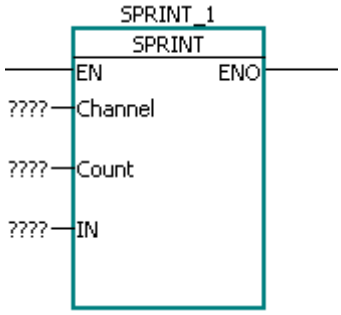
*__OBS__: As seguintes condições devem ser atendidas:
- **YmRa** e **YmRd** devem ser menores que **Ymax**
- **Xmax** deve ser maior que **Xra** + **Xrd**

Caso uma dessas condições não for respeitada, ou o valor da saída for superior ao máximo o flag de overflow é ligado.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	UDINT	Valor Efetivo da Entrada.
	K	BOOL	Habilita Cálculo dos Coeficientes de Ajuste Aa, Ba, Ca, Ad, Bd e Cd.
	YmRa	UINT	Mínimo valor de Y na Rampa de aceleração.
	YmRd		Mínimo valor de Y na Rampa de desaceleração.
	Ymax		Máximo valor de Y.
	Xra		Valor do delta X para rampa de aceleração.
	Xrd		Valor do delta X para rampa de desaceleração.
	Xmax	UDINT	Máximo valor de contagem.
	Zero	BOOL	TRUE – Zera a saída OUT quando os parâmetros de entrada estão incorretos. FALSE – A saída OUT permanece com o último valor calculado caso haja inconsistência nos parâmetros de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	OUT	UINT	Valor da Saída Calculada.
	Aa	REAL	Coeficiente "A" da equação da rampa de aceleração.
	Ba		Coeficiente "B" da equação da rampa de aceleração.
	Ca		Coeficiente "C" da equação da rampa de aceleração.
	Ad		Coeficientes "A" equação da rampa de desaceleração.
	Bd		Coeficientes "B" equação da rampa de desaceleração.
	Cd		Coeficientes "C" equação da rampa de desaceleração.

	FLAG	NOME	DESCRIÇÃO
	Overflow	**OV**	É ligado se houver estouro ou erro no calculo dos coeficientes.
	Zero	**Z**	É ligado se o valor da saída for ZERO.

SPRINT – Escrita em canal serial



A instrução será executada sempre que a entrada **EN** estiver **habilitada**.

Esta instrução quando habilitada envia dados de uma STRING ou Array de bytes pelo canal serial.

Antes de executar a instrução, acionando a entrada EN, é **necessário acionar a variável de sistema** que dispõe o canal serial para enviar valores. Para o **canal A (RS232)** habilitar a variável de sistema **PRINT_A** e para o **canal B (RS485)** habilitar a variável de sistema **PRINT_B**.

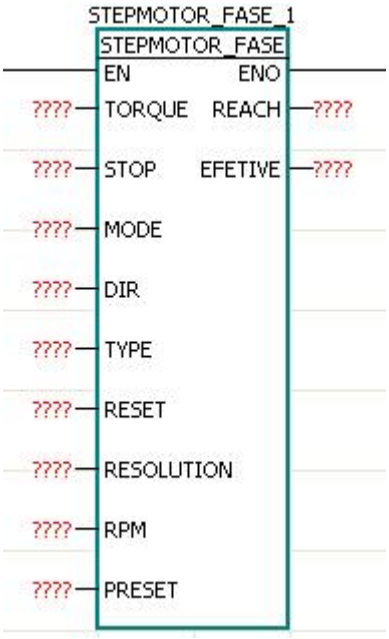
Enquanto a variável de sistema (PRINT_A ou PRINT_B) que habilita o canal serial estiver acionada, o canal de comunicação serial ficará indisponível para supervisão e envio de programa. Desta forma, a variável deve possuir uma lógica associada para ser habilitada somente durante a execução da instrução.

Observação: Utilizando o tipo de dado STRING na entrada IN, as seguintes características são observadas ao executar a instrução:

- Tamanho atual da STRING menor que COUNT: Instrução não é executada. ENO é ligado;
- COUNT maior que o tamanho total da STRING: Instrução não é executada. ENO é ligado;
- Quantidade de caracteres igual ao COUNT: Toda a STRING é enviada pela serial;
- Tamanho atual da STRING maior que COUNT: Será enviado apenas os caracteres definidos até COUNT. Por exemplo: Se COUNT=3 e STRING=ABCD, será enviado apenas os caracteres ABC.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	Channel	UINT	Seleciona canal de comunicação: 0 - RS232 1 - RS485
	Count	UINT	Quantidade de caracteres/bytes a serem transmitidos pelo canal serial.
	IN	STRING Array de byte	Variável contendo dados a serem enviados.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

STEPMOTOR_FASE – Motor de passo modo fase



A **CPU MPC4004.05BF (Tipo N)** permite o acionamento de 1 motor de passo de 4 fases x 2A (corrente máxima por fase), podendo ser ligado diretamente nas saídas do controlador.

O Motor de Passo é controlado pelas 4 saídas S0 a S3, sendo as demais S4 a S7, de uso geral.

Esquema de ligação

- Ver Tipo "N" - 4004.05BF.

Descrição das entradas

TORQUE: Quando ativado (True) energiza o motor com o último passo ativo.

STOP: Quando ativado (True) inibi a progressão de contagem, parando o motor instantaneamente deixando torque no eixo do motor.

MODE

Escolha do modo de funcionamento:

- False - modo posição;
- True - modo contínuo.

Modo contínuo - nesta condição após a habilitação do motor, o mesmo começará a girar indefinidamente. No modo contínuo o motor somente para quando: Desabilitado o torque, ativado stop ou desacionada a instrução.

Modo posição - nesta condição, o motor se deslocará uma quantidade programada de pulsos, parando com torque no final da contagem.

DIR - Direção

Altera o sentido de rotação do eixo. O function block realiza a alteração automaticamente.

- True = Sentido horário;
- False = Sentido anti-horário.

TYPE: Escolha do tipo de passo.

- True = Passo inteiro;
- False = Meio passo.

A opção meio passo permite dobrar a resolução do motor.

- Ver Passo Inteiro;
- Ver Meio Passo.

RESET

Ao ser ativado resetará o efetivo. Usar contato de transição positiva ou negativa.

RESOLUTION

A resolução do motor é especificada pelo fabricante do motor, representa o número de passos que o motor deverá realizar para completar uma volta.

Exemplo:

Para um motor com resolução de 200 (passos por volta), serão necessários duzentos passos para que o eixo do motor complete uma volta.

RPM – Rotações Por Minuto

Valor da velocidade do motor.

A velocidade do motor em RPM calculada difere para cada uma das configurações do motor de passo.

RPM = [60/(Freq. máx. de Interrupção x Resolução)]

Freq. máx. de interrupção = 10 KHz = (1/100 µs) - (Valor constante)

Portanto,

RPM < (600000/Resolução)

A velocidade máxima efetivamente alcançada depende do tipo de motor que se está utilizando, bem como do torque necessário ao processo (Quanto mais veloz menor será o torque do motor).

PRESET

Número de passos que o eixo do motor deverá realizar.

Observação:

Freqüência máxima de interrupção do function block no CLP é 10kHz (máximo).

Período de interrupção = [60 / (RPM x Resolução)].

Utilizando este function block não será permitido adicionar ao programa as seguintes instruções:

- Angle_simulate;
- Stepmotor_pulse.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	TORQUE		Habilita torque no motor.
	STOP		Interrompe funcionamento do motor.
	MODE		False = Modo Posição True = Modo Contínuo
	DIR		False = Sentido anti-horário True = Sentido horário
	TYPE		False = Meio passo True = Passo inteiro
	RESET		Zera o efetivo e desliga a saída REACH se estiver ligada (TRUE)
	RESOLUTION	UDINT	Resolução (valor específico do motor de passo utilizado).
	RPM		Rotações por minuto.
	PRESET		Número de passos do motor.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	REACH		Liga quando Efetive=Preset.
	EFETIVE	UDINT	Número de passos dado pelo motor.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

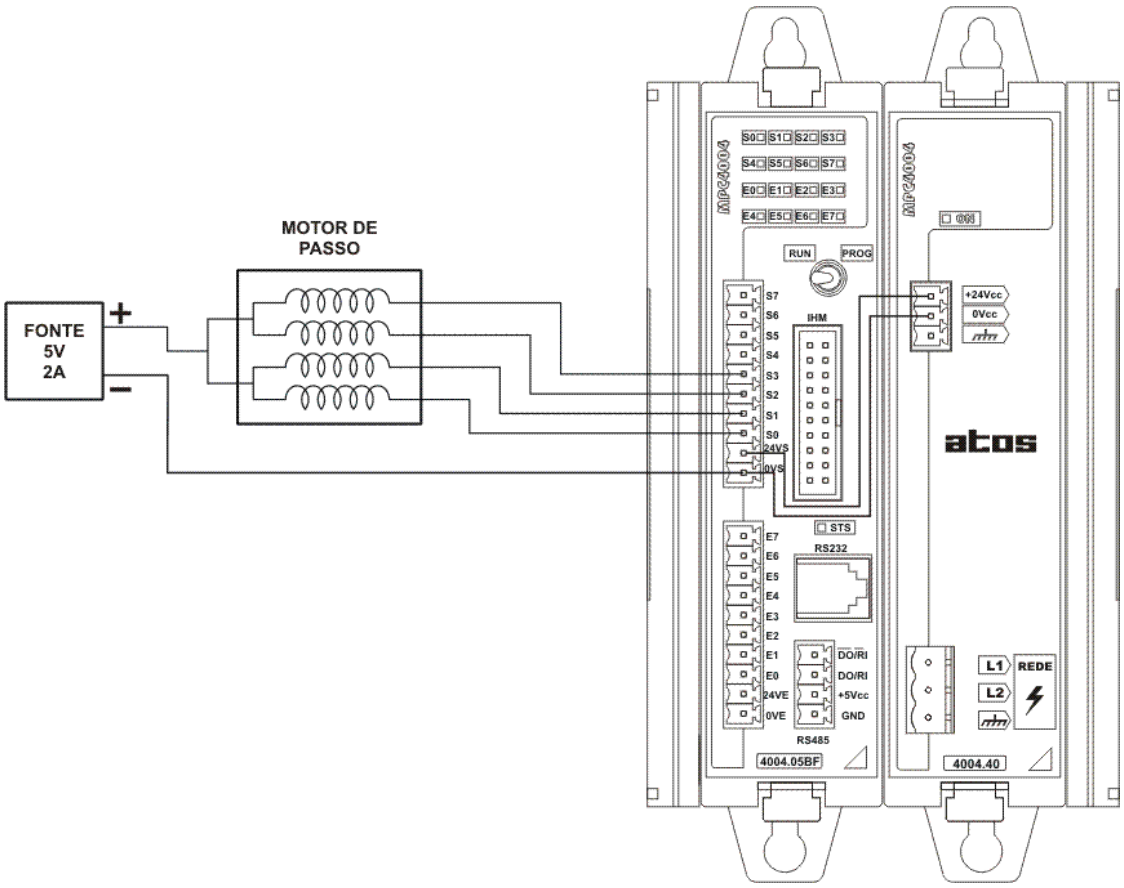
Passo inteiro

	S0	S1	S2	S3
1	True	False	True	False
2	True	False	False	True
3	False	True	False	True
4	False	True	True	False

Meio Passo

	S0	S1	S2	S3
0.5	True	True	True	False
1.0	True	False	True	False
1.5	True	False	True	True
2.0	True	False	False	True
2.5	True	True	False	True
3.0	False	True	False	True
3.5	False	True	True	True
4.0	False	True	True	False

Tipo "N" - 4004.05BF (Somente para tipo N)



As saídas do controlador programável podem ser divididas em dois módulos:

Circuito de controle: necessita de uma tensão de 24V/10mA para funcionar.

Circuito de potência: pode ser considerado um circuito com coletor aberto, permitindo a conexão de tensões que podem variar de 3 a 30V com correntes de até 2A.

Desta forma, uma vez polarizado o circuito de controle em 24V, a ligação do motor de passo nas saídas poderá ser feito com tensões mais baixas, por exemplo, 5V sem a necessidade de limitadores de tensão.

STEPMOTOR_PULSE – Motor de passo modo pulse

As CPUs **MPC4004.05BF** e **MPC4004.06BF** permitem o acionamento de motor de passo de 4 fases x 2 A (corrente máxima por fase), podendo gerar pulsos para alimentar a entrada de um driver de motor de passo.

A **saída S0** gera pulsos para o driver do motor de passo, sendo as demais S1 a S7 de uso geral.

Esquema de ligação

- Ver Tipo "N" - 4004.05BF;
- Ver Tipo "P" - 4004.06BF.

Descrição das entradas

BLOCK: Quando ativado (True) inibi a saída de pulsos, parando o motor instantaneamente.

MODE

Escolha do modo de funcionamento:

- False - modo posição;
- True - modo contínuo.

Modo contínuo - nesta condição após a habilitação do motor, o mesmo começará a girar indefinidamente. No modo contínuo o motor somente para quando: Desabilitado o torque, ativado stop ou desacionada a instrução.

Modo posição - nesta condição, o motor se deslocará uma quantidade programada de pulsos, parando com torque no final da contagem.

RESET

Ao ser ativado resetará o efetivo. Usar contato de transição positiva ou negativa.

RESOLUTION

A resolução do motor é especificada pelo fabricante do motor, representa o número de passos que o motor deverá realizar para completar uma volta.

Exemplo:

Para um motor com resolução de 200 (passos por volta), serão necessários duzentos passos para que o eixo do motor complete uma volta.

RPM – Rotações Por Minuto

Valor da velocidade do motor.

A velocidade do motor em RPM calculada difere para cada uma das configurações do motor de passo.

RPM = [60 / (Freq. máx. de Interrupção x Resolução)]

Freq. máx. de interrupção = 10 KHz = (1/100 µs) - (Valor constante)

Portanto,

RPM < (600000 / Resolução)

A velocidade máxima efetivamente alcançada depende do tipo de motor que se está utilizando, bem como do torque necessário ao processo (Quanto mais veloz menor será o torque do motor).

PRESET

Número de passos que o eixo do motor deverá realizar.

Observação:

Freqüência máxima de interrupção do function block no CLP é 10kHz (máximo).

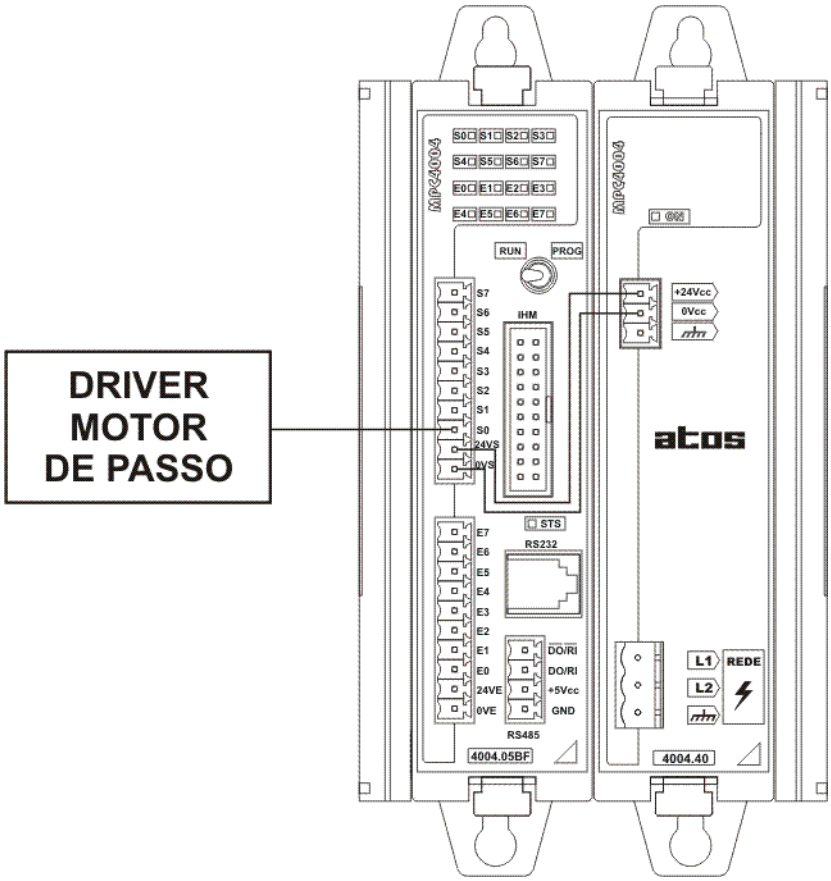
Período de interrupção = [60 / (RPM x Resolução)].

Utilizando este function block não será permitido adicionar ao programa as seguintes instruções:

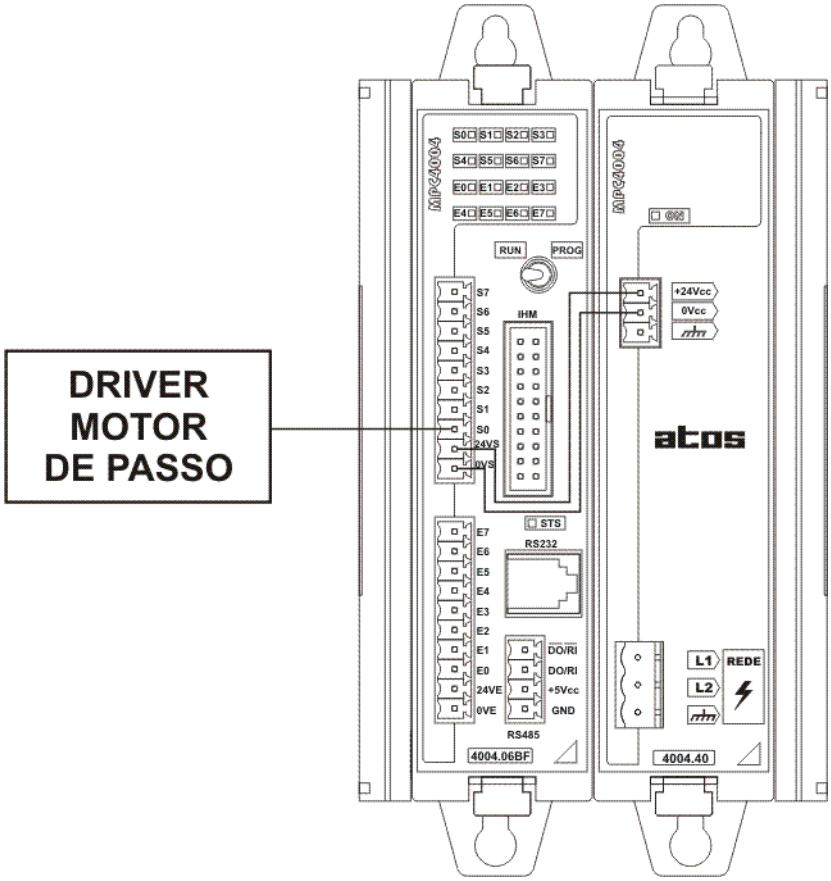
- Angle_simulate;
- Stepmotor_fase.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	BLOCK		Interrompe funcionamento do motor.
	MODE		False = Modo Posição True = Modo Contínuo
	RESET		Zera o efetivo e desliga a saídaREACH se estiver ligada (TRUE)
	RESOLUTION	UDINT	Resolução (valor específico do motor de passo utilizado).
	RPM		Rotações por minuto.
	PRESET		Número de passos do motor.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	REACH		Liga quando Efetive=Preset.
	EFETIVE	UDINT	Número de passos dado pelo motor.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Tipo "N" - 4004.05BF

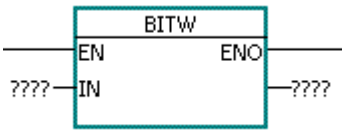


Tipo "P" - 4004.06BF



Movimentação de dados

BITW - Movimentação BIT >>> WORD



Movimentação de estado para palavra.

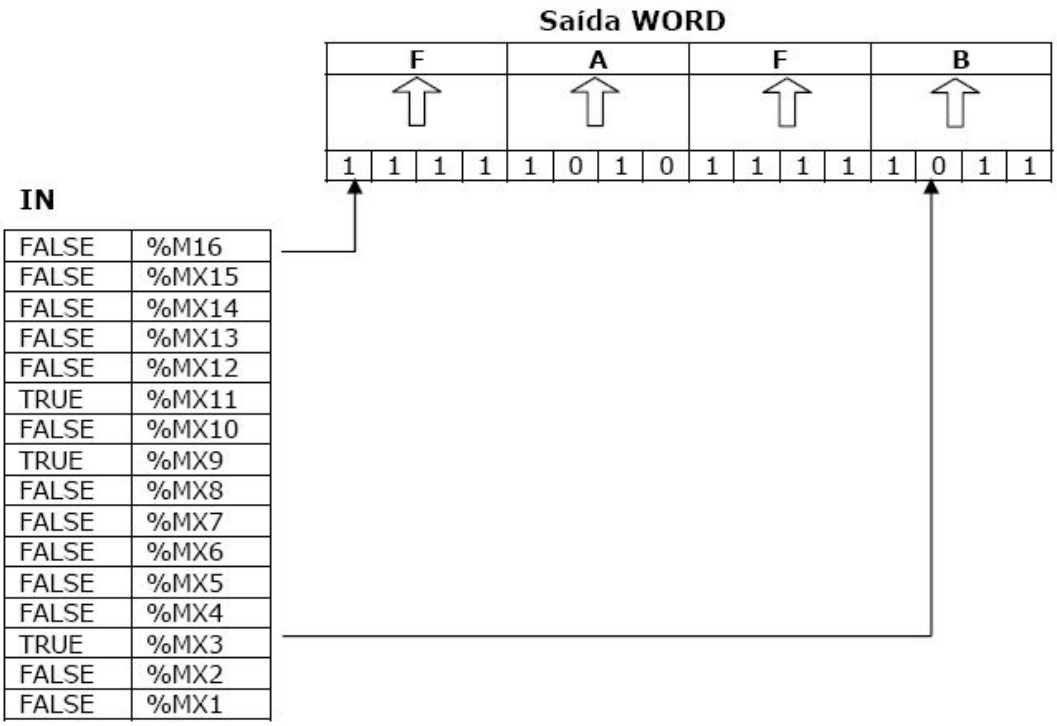
Quando a entrada **EN** for habilitada, esta instrução executa a transferência de "**N**" variáveis booleanas consecutivas, apontadas a partir da variável **IN**, para o conteúdo da variável declarada na saída da função, onde "**N**" será igual ao tamanho da variável declarada na saída, se Word (16 bits) ou Dword (32 bits).

ENO será uma cópia de EN.

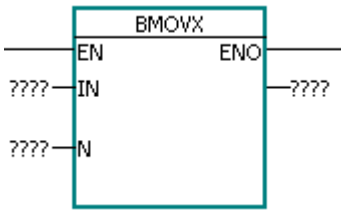
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	BOOL	Índice para variáveis booleanas.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	WORD e DWORD	Variável de saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Exemplo de BITW:

Declarando uma variável do tipo Word (16 bits) para a saída, teremos a partir da variável IN %MX1, 16 variáveis booleanas de entrada (%MX1 a %MX16) que serão transferidas para word.



BMOVX - Movimentação indexada na origem e destino



Esta instrução copia uma determinada quantidade de dados **N**, a partir do endereço apontado em IN (endereço de origem), para uma outra região que começa no endereço apontado pela **saída da função** (endereço de destino).
A entrada **N** define o número de variáveis a serem movimentadas. Observe que a quantidade de bytes movida depende o tipo de dado utilizado.

Utilizando o tipo de dado WORD em IN e na **saída da função**, move-se **N** dados de 16 bits.
Utilizando o tipo de dado DWORD em IN e na **saída da função**, move-se **N** dados de 32 bits.

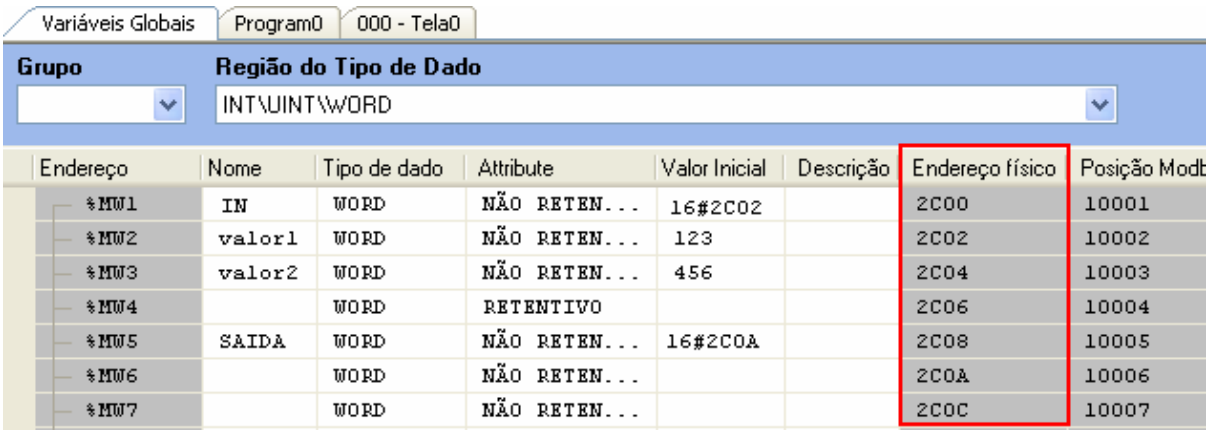
Atenção: O endereço definido em IN e na saída da função é o endereço físico da origem e do destino do dado.

Observações: A instrução BMOVX pode ler toda a memória do usuário podendo escrever em qualquer região da memória que não seja a região de sistema e I/O.
Utilize o mesmo tipo de dado nas regiões apontadas por IN e OUT, para garantir que os dados sejam corretamente interpretados pelo CLP.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	IN	WORD e DWORD	Variável de entrada
	N	UINT	Quantidade de dados
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	Função	WORD e DWORD	Ponteiro para saída
	FLAG	**NOME**	**DESCRIÇÃO**
	Overflow	**OV**	Será ligado se a variável usada sair do **range de endereços** do tipo de dado.

Exemplo de BMOVX:

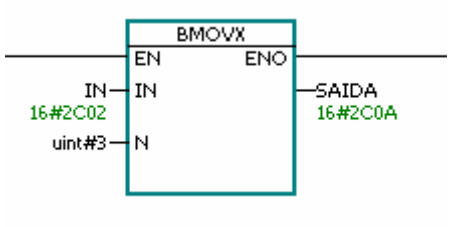
O endereço definido em IN e na saída da função é o endereço físico, mostrado na tabela do gerenciador de variáveis, conforme a figura abaixo:


Variáveis Globais | Program0 | 000 - Tela0

Grupo: — Região do Tipo de Dado: INT\UINT\WORD

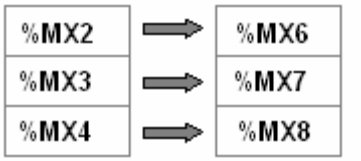
Endereço	Nome	Tipo de dado	Attribute	Valor Inicial	Descrição	Endereço físico	Posição Modb
%MW1	IN	WORD	NÃO RETEN...	16#2C02		2C00	10001
%MW2	valor1	WORD	NÃO RETEN...	123		2C02	10002
%MW3	valor2	WORD	NÃO RETEN...	456		2C04	10003
%MW4		WORD	RETENTIVO			2C06	10004
%MW5	SAIDA	WORD	NÃO RETEN...	16#2C0A		2C08	10005
%MW6		WORD	NÃO RETEN...			2C0A	10006
%MW7		WORD	NÃO RETEN...			2C0C	10007

Deseja-se mover o conteúdo de %MW2 para variável %MW6.
O exemplo utiliza as variáveis IN e SAIDA na instrução.
O conteúdo da variável IN deve ser o endereço físico da variável a ser copiada e o conteúdo da variável SAIDA, o endereço físico da variável de destino.

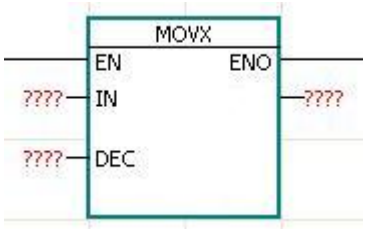


No nosso caso, desejamos copiar o conteúdo de %MW2, de endereço físico 16#2C02 e copiar para variável %MW6, de endereço físico 16#2C0A. Para isso inicializamos as variáveis conforme a tabela acima.

Dessa forma, ao habilitar o bloco, o conteúdo de %MW2 será copiado para %MW6, como para N está definida a constante 3, as três variáveis na sequência também serão copiadas, conforme mostra a tabela abaixo:



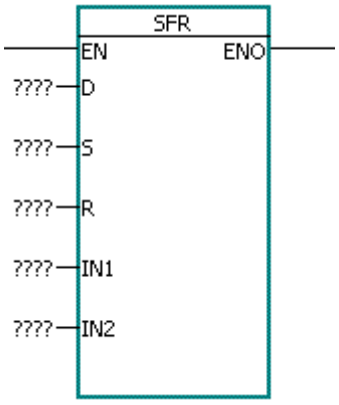
MOVX - Movimentação indexada no destino



Se a entrada EN estiver acionada o conteúdo da variável IN será transferido para o a variável cujo índice (endereço) é apontado na **saída da função**.
A instrução tem ainda uma entrada DEC (decremento), que quando ativada provoca um decremento automático no índice (**saída da função**), caso contrário provoca um incremento.
Como o incremento ou decremento é feito antes da execução da transferência, é necessário que o conteúdo inicial do índice seja uma posição de memória a menos ou a mais conforme a operação seja de incremento ou decremento respectivamente.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	INT, DINT, UINT, DINT, WORD, DWORD, TIME, DATE, TOD, REAL, e CONSTANTE.	Variável de entrada.
	DEC	BOOL	Incrementa/decrementa ponteiro de destino.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função	WORD e DWORD	Ponteiro para inserir IN.
	FLAG	**NOME**	**DESCRIÇÃO**
	Overflow	**OV**	Será ligado se a variável usada sair do **range de endereços** do tipo de dado.

SFR - Deslocamento

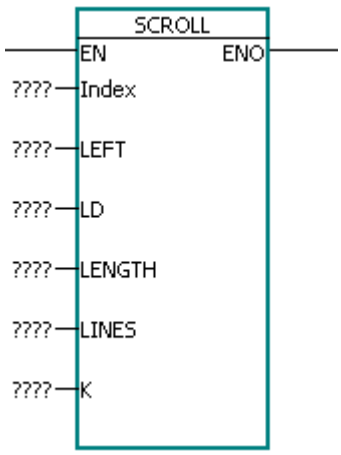


Esta é uma instrução de deslocamento. Funciona como uma entrada serial a um registro de deslocamento. Se a entrada EN estiver ligada, a instrução se comporta de seguinte maneira:

- Se a entrada **R** estiver habilitada, todas as variáveis booleanas entre as entradas **IN1** e **IN2** serão desligadas. Se a mesma estiver desabilitada, será feito o deslocamento das variáveis booleanas.
- Se a entrada **S** estiver **desabilitada**, será feito um deslocamento de **IN1 para IN2** e o conteúdo da entrada **D** será **copiado** para variável booleana **IN1**. Caso a entrada **S** esteja **habilitada**, será feito um deslocamento de **IN2 para IN1** e o conteúdo da entrada **D** será **copiado** para variável booleana **IN2**.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Utilizar **pulsos** para a execução da instrução.
	D		Variável de entrada dos dados.
	S		Indica se o deslocamento é no sentido de IN1 para IN2 (**S desacionado**) ou IN2 para IN1 (**S acionado**).
	R		Habilita Desligamento das variáveis booleanas dentro da faixa entre IN1 e IN2.
	IN1, IN2		Variáveis booleanas que definem a faixa.
SAÍDA	ENO		Cópia do valor booleano de EN.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	**-----**	Nenhum *flag* é afetado.

SCROLL – Deslocamento de bloco de dados



A instrução será executada sempre que a entrada EN estiver habilitada por pulsos de transição positiva.

Esta instrução quando habilitada executa movimentação de blocos de dados de acordo com as entradas.

Index: Primeira variável de uma lista.

LENGHT: Tamanho do bloco de dados a ser movimentado.

LINES: Quantidade de variáveis listadas incluindo a variável index.

K: Constante copiada para todas as variáveis da lista. Deve ser o mesmo tipo de dado das variáveis listadas.

Ver Exemplo 1

Movimentação de dados da esquerda para direita

LEFT = FALSE e LD = FALSE

Com as entradas LEFT e LD em FALSE a cada pulso na entrada **EN**, o conteúdo da variável **Index** é copiado para a posição: **Index+LENGHT**, descartando os dados das variáveis no final da lista.

Ver Exemplo 2

Movimentação de dados da direita para esquerda

LEFT = TRUE e LD = FALSE

A cada pulso na entrada EN, o conteúdo da última variável da lista é copiado para a posição: última variável da lista-LENGHT, descartando os dados das variáveis no início da lista.

Ver Exemplo 3

Carregar valor de uma constante

LEFT = TRUE e LD = TRUE
ou
LEFT = FALSE e LD = TRUE

Aplicando um pulso na entrada **EN**, o valor de **K** é copiado para o conteúdo de todas as variáveis da lista.

Ver Exemplo 4

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Deve ser acionada por pulso de transição positiva.
	Index	INT, DINT, UINT, DINT, WORD, DWORD, REAL, TIME, DATE, DATE_AND_TIME e TOD.	Entrada com valor a ser copiado.
	LEFT	BOOL	Deslocamento de dados para direita ou para esquerda.
	LD		Copia o valor de K para todas as variáveis listadas.
	LENGTH	CONSTANTE do tipo UINT	Tamanho do bloco de variáveis.
	LINES		Quantidade de variáveis listadas.
	K	CONSTANTE do tipo: INT, DINT, UINT, UDINT, WORD, DWORD, REAL, TIME, DATE, DATE_AND_TIME e TOD.	Constante a ser copiada para o conteúdo de todas as variáveis listadas.
SAÍDA	ENO	BOLL	Cópia do valor booleano de EN.
	FLAG	NOME	DESCRIÇÃO
	-----	-----	Nenhum *flag* é afetado.

Exemplo 1

Entradas configuradas:

- **Index** = %MW1 (variável do tipo WORD);
- **LENGHT** = UINT#2 (constante de valor 2 do tipo UINT);
- **LINES** = UINT#10 (constante de valor 10 do tipo UINT).

	ENDEREÇO	DADO
Variáveis Listadas	%MW1	11
	%MW2	22
	%MW3	33
	%MW4	44
	%MW5	55
	%MW6	66
	%MW7	77
	%MW8	88
	%MW9	99
	%MW10	100

Exemplo 2

Entradas configuradas:

- **Index = %MW1** (variável do tipo WORD);
- **LENGHT = UINT#2** (constante de valor **2** do tipo UINT);
- **LINES = UINT#10** (constante de valor 10 do tipo UINT).

	ENDEREÇO	DADO	Pulso1	Pulso2	Pulso3	Pulso4	Pulso5
Variáveis Listadas	%MW1	11	11	11	11	11	
	%MW2	22	22	22	22	22	
	%MW3	33	11	11	11	11	
	%MW4	44	22	22	22	22	
	%MW5	55	33	11	11	11	
	%MW6	66	44	22	22	22	
	%MW7	77	55	33	11	11	
	%MW8	88	66	44	22	22	
	%MW9	99	77	55	33	11	
	%MW10	100	88	66	44	22	

Entradas configuradas:

- **Index = %MW1** (variável do tipo WORD);
- **LENGHT = UINT#5** (constante de valor **5** do tipo UINT);
- **LINES = : UINT#10** (constante de valor 10 do tipo UINT).

Antes do segundo pulso é inserido um novo dado (99) na variável Index.
Antes do quarto pulso é inserido um novo dado (1000) na variável Index.

	ENDEREÇO	DADO	Pulso1	Pulso2	Pulso3	Pulso4	Pulso5
Variáveis Listadas	%MW1	11	11	99	99	1000	1000
	%MW2	22	22	22	22	22	22
	%MW3	33	33	33	33	33	33
	%MW4	44	44	44	44	44	44
	%MW5	55	55	55	55	55	55
	%MW6	66	11	11	99	99	1000
	%MW7	77	22	22	22	22	22
	%MW8	88	33	33	33	33	33
	%MW9	99	44	44	44	44	44
	%MW10	100	55	55	55	55	55

Exemplo 3

Entradas configuradas:

- **Index = %MW1** (variável do tipo WORD);
- **LENGHT = UINT#1** (constante de valor **1** do tipo UINT);
- **LINES = UINT#10** (constante de valor 10 do tipo UINT).

	ENDEREÇO	DADO	Pulso1	Pulso2	Pulso3	Pulso4	Pulso5
Variáveis Listadas	%MW1	11	22	33	44	55	66
	%MW2	22	33	44	55	66	77
	%MW3	33	44	55	66	77	88
	%MW4	44	55	66	77	88	99
	%MW5	55	66	77	88	99	100
	%MW6	66	77	88	99	100	100
	%MW7	77	88	99	100	100	100
	%MW8	88	99	100	100	100	100
	%MW9	99	100	100	100	100	100
	%MW10	100	100	100	100	100	100

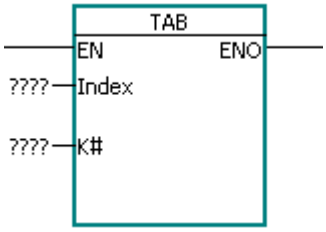
Exemplo 4

Entradas configuradas:

- **Index = %MW1** (variável do tipo WORD);
- **LENGHT = UINT#2** (constante de valor 2 do tipo UINT);
- **LINES = UINT#10** (constante de valor 10 do tipo UINT);
- **K = WORD#200** (constante de valor **200** do tipo WORD);
- **LD = TRUE**.

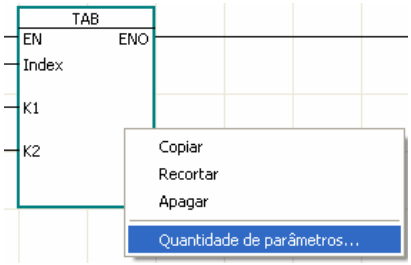
	ENDEREÇO	DADO	Pulso1
Variáveis Listadas	%MW1	11	200
	%MW2	22	200
	%MW3	33	200
	%MW4	44	200
	%MW5	55	200
	%MW6	66	200
	%MW7	77	200
	%MW8	88	200
	%MW9	99	200
	%MW10	100	200

TAB - Carregamento de constantes

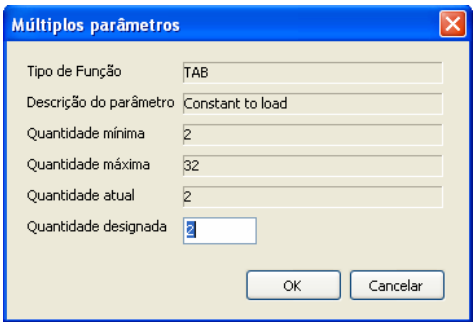


Esta função carrega um conjunto de constantes a partir de um endereço especificado na entrada "Index".

A quantidade de entradas de constantes do bloco é configurável via software (2 a 32 entradas). Essa configuração é feita clicando com o botão direito sobre o bloco TAB. No menu pop-up que aparecer, selecione a opção "*Quantidade de parâmetros*".

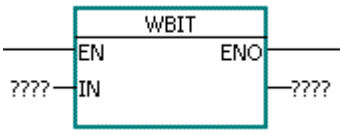


Na janela de propriedades do bloco, defina a quantidade de parâmetros que o bloco vai possuir.



	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução
	Index	INT, DINT, UINT, UDINT, WORD, DWORD, TIME, DATE, TOD e REAL.	Ponteiro de destino das constantes K#
	K#	CONSTANTE	Constantes a serem carregadas a partir de 'Index'.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

WBIT - Movimentação WORD >>> BIT

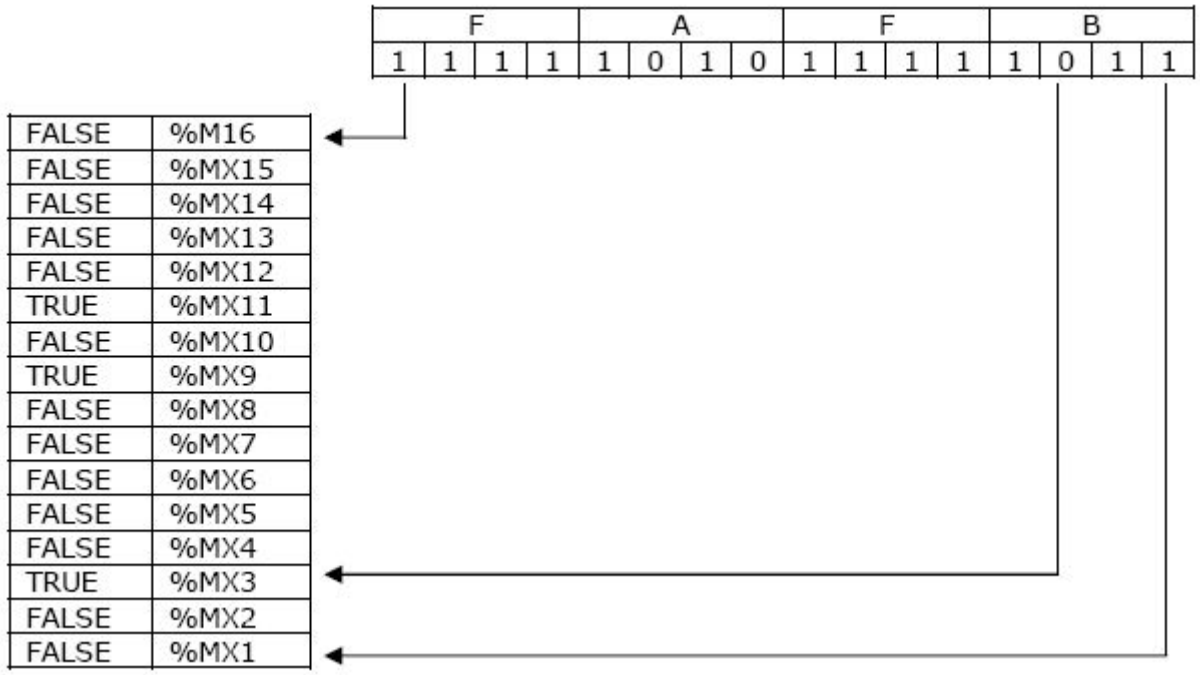


Quando a entrada EN for habilitada, o conteúdo da variável IN será movido para "N" variáveis booleanas apontadas pela variável da **saída da função**. "N" será igual ao tamanho da variável IN. EN0 será uma cópia de EN. O bit "0" de IN será o conteúdo da primeira booleana.

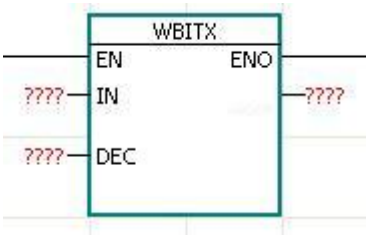
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	WORD e DWORD.	Variável de entrada.
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN.
	Função		Índice para variável de saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Exemplo de WBIT:

Caso o conteúdo da variável IN for 16#FAFB e a saída da função for %MX1, as variáveis booleanas %MX1 a %MX16 serão afetadas da seguinte forma:



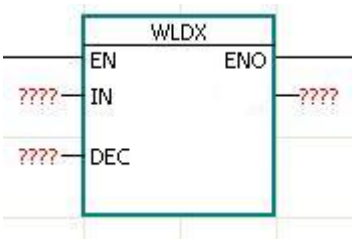
WBITX - Movimentação indexada WORD >>> BIT



Quando a entrada EN for habilitada, o endereço apontado pela variável IN será incrementado ou decrementado, conforme o estado da entrada I/D e o conteúdo deste endereço movido para "N" variáveis booleanas apontadas pela variável da **saída da função**. "N" será igual ao tamanho da variável IN. EN0 será uma cópia de EN. O bit "0" de IN será o conteúdo da primeira booleana.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	WORD e DWORD	Ponteiro para a Variável de Entrada.
	DEC	BOOL	Quando ativada provoca um decremento automático, caso contrário um incremento automático do Endereço apontado pela Variável IN.
SAÍDA	ENO		Cópia do valor booleano de EN.
	Função		Variável Booleana Inicial de Saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	Overflow	**OV**	Será ligado se a variável usada sair do **range de endereços** do tipo de dado.

WLDX - Movimentação indexada na origem



Quando a entrada EN for habilitada, o conteúdo do endereço apontado pela variável IN será incrementado ou decrementado, conforme o estado da entrada DEC e o conteúdo deste endereço será movido para a variável da **saída da função**. EN0 será uma cópia de EN.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução.
	IN	WORD e DWORD	Ponteiro para a Variável de Entrada.
	DEC	BOOL	Quando ativada provoca um decremento automático, caso contrário um incremento automático do Endereço apontado pela Variável IN.
SAÍDA	ENO		Cópia do valor booleano de EN.
	Função	INT, DINT, UINT, UDINT, WORD, DWORD, TIME, DATE, TOD, REAL e CONSTANTE.	Variável de Saída.
	FLAG	**NOME**	**DESCRIÇÃO**
	Overflow	**OV**	Será ligado se a variável usada sair do **range de endereços** do tipo de dado.

Exemplo de range de dados

Para um range de variáveis de 16 bits já definido pelo usuário, pode-se verificar na região de memória do item **"Variáveis Globais"** o valor de **endereço mínimo** e o valor de **endereço máximo**. Se o decremento ultrapassar o endereço mínimo o Overflow será acionado e para um incremento que ultrapasse o endereço máximo o Overflow também será acionado.

Endereço Mínimo

Variáveis Globais

Grupo USUÁRIO | **Região do Tipo de Dado** INT\UINT\WORD

Endereço	Nome	Tipo de dado	Attribute	Valor Inicial	Descrição	Endereço físico	Posição Modbus
%MW1		INT	RETENTIVO			2C00	10001
%MW2		INT	RETENTIVO			2C02	10002
%MW3		INT	RETENTIVO			2C04	10003
%MW4		INT	RETENTIVO			2C06	10004
%MW5		INT	RETENTIVO			2C08	10005
%MW6		INT	RETENTIVO			2C0A	10006
%MW7		INT	RETENTIVO			2C0C	10007
%MW8		INT	RETENTIVO			2C0E	10008
%MW9		INT	RETENTIVO			2C10	10009
%MW10		INT	RETENTIVO			2C12	10010
%MW11		INT	RETENTIVO			2C14	10011
%MW12		INT	RETENTIVO			2C16	10012
%MW13		INT	RETENTIVO			2C18	10013
%MW14		INT	RETENTIVO			2C1A	10014
%MW15		INT	RETENTIVO			2C1C	10015
%MW16		INT	RETENTIVO			2C1E	10016

Endereço Máximo

Variáveis Globais

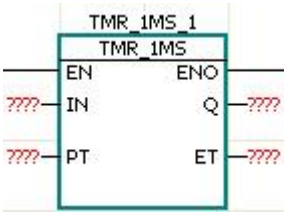
Grupo USUÁRIO | **Região do Tipo de Dado** INT\UINT\WORD

Endereço	Nome	Tipo de dado	Attribute	Valor Inicial	Descrição	Endereço físico	Posição Modbus
%MW2033		INT	RETENTIVO			3BE0	12033
%MW2034		INT	RETENTIVO			3BE2	12034
%MW2035		INT	RETENTIVO			3BE4	12035
%MW2036		INT	RETENTIVO			3BE6	12036
%MW2037		INT	RETENTIVO			3BE8	12037
%MW2038		INT	RETENTIVO			3BEA	12038
%MW2039		INT	RETENTIVO			3BEC	12039
%MW2040		INT	RETENTIVO			3BEE	12040
%MW2041		INT	RETENTIVO			3BF0	12041
%MW2042		INT	RETENTIVO			3BF2	12042
%MW2043		INT	RETENTIVO			3BF4	12043
%MW2044		INT	RETENTIVO			3BF6	12044
%MW2045		INT	RETENTIVO			3BF8	12045
%MW2046		INT	RETENTIVO			3BFA	12046
%MW2047		INT	RETENTIVO			3BFC	12047
%MW2048		INT	RETENTIVO			3BFE	12048

Temporização e Contagem

TMR_1MS - Temporizador: 1ms OFF Delay

O Function Block TMR_1MS pode ser utilizado de dois modos distintos: Com e sem EN/ENO. Cada modo possui uma característica diferente de trabalho e podem ser utilizados individualmente para cada TMR_1MS inserido no programa. **O projeto pode ter até 2 TMR_1MS.**



Modo: Com EN/ENO

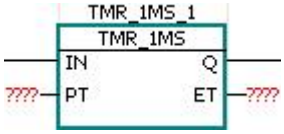
Com as entradas EN e IN habilitadas, inicia-se a contagem de tempo na **saída ET (Base de tempo 1ms)** até o valor de preset (PT). Durante essa contagem, a saída Q permanece ligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será ligada e permanecerá assim até que IN seja desligado. Quando desligado, a saída Q é desligada e o efetivo ET zerado (ET= 0).

Se EN for desligado antes do fim da temporização, a temporização entra em PAUSE, voltando ao normal quando EN for novamente habilitado.

O RESET de temporização acontece quando a entrada IN for desligada, somente se EN estiver habilitado. Do contrário o bloco não tem nenhum status alterado.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q ligada.



Modo: Sem EN/ENO

Com a entrada IN habilitada, inicia-se a contagem de tempo na **saída ET (Base de tempo 1ms)** até o valor de preset (PT). Durante essa contagem, a saída Q permanece ligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será ligada e permanecerá assim até que IN seja desligado. Quando desligado, a saída Q é desligada e o efetivo ET zerado (ET= 0).

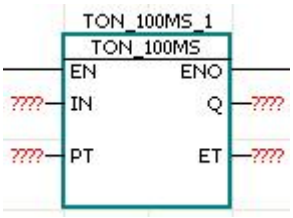
O RESET de temporização acontece quando a entrada IN for desligada, somente se EN estiver habilitado. Do contrário o bloco não tem nenhum status alterado.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q ligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Pausa temporização ao desabilitar antes do fim da contagem de tempo. Entrada **opcional**. Existente somente no modo com EN/ENO.
	IN	BOOL	Iniciar temporização
	PT	TIME e CONSTANTE	Preset do temporizador
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Saída **opcional**. Existente somente no modo com EN/ENO.
	Q	BOOL	Habilitado no fim da temporização
	ET	TIME	Efetivo do temporizador
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

TON_100MS - Temporizador: 100ms ON Delay

O Bloco Funcional TON_100MS pode ser utilizado de dois modos distintos: Com e sem EN/ENO. Cada modo possui uma característica diferente de trabalho e podem ser utilizados individualmente para cada TON_100MS inserido no programa. **O projeto pode ter até 1536 TON_100MS.**



Modo: Com EN/ENO

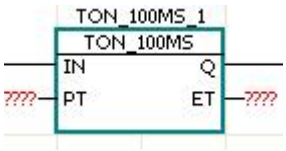
Com as entradas EN e IN habilitadas, inicia-se a contagem de tempo na **saída ET (Base de tempo 100ms)** até o valor de preset (PT). Durante essa contagem, a saída Q permanece desligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será ligada e permanecerá assim até que IN seja desligado. Quando desligado, a saída Q é desligada e o efetivo ET zerado (ET= 0).

Se EN for desligado antes do fim da temporização, a temporização entra em PAUSE, voltando ao normal quando EN for novamente habilitado.

O RESET de temporização acontece quando a entrada IN for desligada, somente se EN estiver habilitado. Do contrário o bloco não tem nenhum status alterado.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q ligada.



Modo: Sem EN/ENO

Com a entrada IN habilitada, inicia-se a contagem de tempo na **saída ET (Base de tempo 100ms)** até o valor de preset (PT). Durante essa contagem, a saída Q permanece desligada.

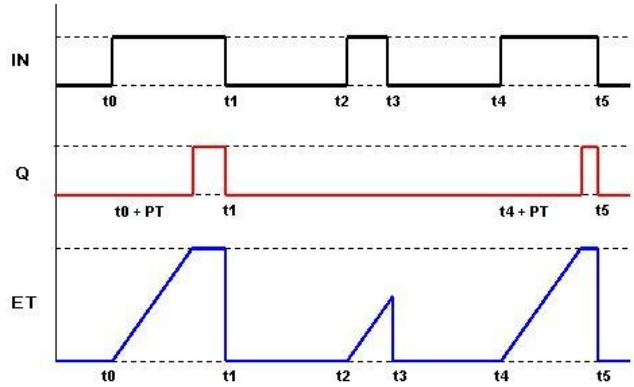
Quando o efetivo for igual ao preset (ET=PT), a saída Q será ligada e permanecerá assim até que IN seja desligado. Quando desligado, a saída Q é desligada e o efetivo ET zerado (ET= 0).

O RESET de temporização acontece quando a entrada IN for desligada, somente se EN estiver habilitado. Do contrário o bloco não tem nenhum status alterado.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q ligada.

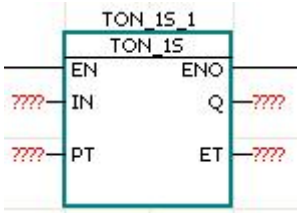
	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Pausa temporização ao desabilitar antes do fim da contagem de tempo. Entrada **opcional**. Existente somente no modo com EN/ENO.
	IN	BOOL	Iniciar temporização
	PT	TIME e CONSTANTE	Preset do temporizador
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Saída **opcional**. Existente somente no modo com EN/ENO.
	Q	BOOL	Habilitado no fim da temporização
	ET	TIME	Efetivo do temporizador
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado.

Exemplo gráfico de funcionamento:



TON_1S - Temporizador: 1s ON Delay

O Function Block TON_1S pode ser utilizado de dois modos distintos: Com e sem EN/ENO. Cada modo possui uma característica diferente de trabalho e podem ser utilizados individualmente para cada TON_1S inserido no programa. **O projeto pode ter até 1536 TON_1S.**



Modo: Com EN/ENO

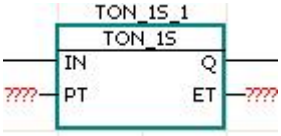
Com as entradas EN e IN habilitadas, inicia-se a contagem de tempo na **saída ET (Base de tempo 1s)** até o valor de preset (PT). Durante essa contagem, a saída Q permanece desligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será ligada e permanecerá assim até que IN seja desligado. Quando desligado, a saída Q é desligada e o efetivo ET zerado (ET= 0).

Se EN for desligado antes do fim da temporização, a temporização entra em PAUSE, voltando ao normal quando EN for novamente habilitado.

O RESET de temporização acontece quando a entrada IN for desligada, somente se EN estiver habilitado. Do contrário o bloco não tem nenhum status alterado.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a saída Q ligada.



Modo: Sem EN/ENO

Com a entrada IN habilitada, inicia-se a contagem de tempo na **saída ET (Base de tempo 1s)** até o valor de preset (PT). Durante essa contagem, a saída Q permanece desligada.

Quando o efetivo for igual ao preset (ET=PT), a saída Q será ligada e permanecerá assim até que IN seja desligado. Quando desligado, a saída Q é desligada e o efetivo ET zerado (ET= 0).

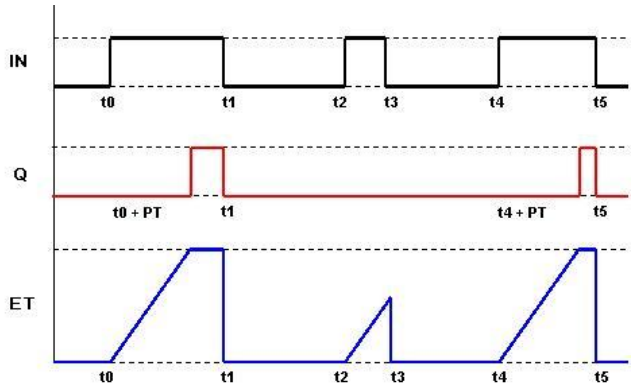
O RESET de temporização acontece quando a entrada IN for desligada, somente se EN estiver habilitado. Do contrário o bloco não tem nenhum status alterado.

O valor do preset PT pode ser alterado durante a execução do bloco funcional. Sendo o valor de PT maior que ET a temporização permanece até ET=PT. Caso contrário, PT menor que ET, a temporização é finalizada e a

saída Q ligada.

	NOME	TIPO DE DADO	DESCRIÇÃO
ENTRADA	EN	BOOL	Habilita execução da instrução. Pausa temporização ao desabilitar antes do fim da contagem de tempo. Entrada **opcional**. Existente somente no modo com EN/ENO.
	IN	BOOL	Iniciar temporização
	PT	TIME e CONSTANTE	Preset do temporizador
SAÍDA	ENO	BOOL	Cópia do valor booleano de EN. Saída **opcional**. Existente somente no modo com EN/ENO.
	Q	BOOL	Habilitado no fim da temporização
	ET	TIME	Efetivo do temporizador
	FLAG	**NOME**	**DESCRIÇÃO**
	-----	-----	Nenhum *flag* é afetado

Exemplo gráfico de funcionamento:



Índice

*****

*_TO_** 34
*_TO_AR 70
*_TO_BCD 34
*_TO_STR 35

A

ABS 25
ACOS 64
ADD 25
ADJUST 77
ALARM 71
AND 38
ANGLE COUNTER 78
ANGLE SIMULATE 79
AR_TO_* 70
ASIN 64
ATAN 65

B

BCD_TO_** 35
BITW 105
BMOVX 106

C

CEP 80
CONCAT 42
Contato de Transição Negativa 20
Contato de Transição Positiva 19
Contato Normalmente Aberto (NA) 19
Contato Normalmente Fechado (NF) 19
COS 65
CPU COUNTER 83
CTD 51
CTU 53
CTUD 55

D

DATE_FORMAT 72
DEG_TO_RAD 73
DELETE 42
DIV 26

E

EQ 31
EVENT_LOG 84
EXP 26
EXPT 27

F

F_TRIG 75
FILT 86
FIND 43

G

GE 31
GT 32

H

HIGH SPEED COUNTER 88

I

INSERT 43

L

LE 32
LEFT 44
LEN 44
LIMIT 47
LN 37
LOG 37
LT 33

M

MAX 48
MID 45
MIN 48
MOD 27
MOVE 28
MOVX 108
MUL 28
MUX 49

N

NE 33
NOT 38

O

OR 39

P

PID 89

R

R_TRIG 76
RAD_TO_DEG 73
RECIPE 91
REPLACE 45
Reset Saída 21
RIGHTLEFT 46
ROL 40
ROR 40
RPRINT 93
RS 30
RTC_ATOS 74

S

Saída 20
Saída de Transição Negativa 21
Saída de Transição Positiva 21
Saída Invertida 20
Saída SET RESET 22
SCALE 95
SCALE2G 96
SCROLL 109
SEL 50
Set Saída 20
SFR 108
SHL 41
SHR 41
SIN 66
SPRINT 97
SQRT 29
SR 30
STEPMOTOR_FASE 98
STEPMOTOR_PULSE 102
STR_TO_** 35
SUB 29

T

TAB 113
TAN 66
TMR_1MS 118
TOF 58
TON 60
TON_100MS 119
TON_1S 120
TP 62

TRUNC 36

W

WBIT 114
WBITX.....115
WLDX 116

X

XOR 39
XOR_AR.....69